Aalto University

School of Science

Degree Programme in Computer Science and Engineering

Emmi Kaivosoja

The Role of Usability in Software Validation - Case: Medical Device Manufacturing

Master's Thesis

Espoo, October 1, 2015

Supervisor: Professor Marko Nieminen, Dr.Sc. (Tech)

Thesis advisor: Ninna Koivunen, M.Sc.

Aalto University
School of Science

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi

Abstract of master's thesis

Author Emmi Kaivosoja

Title of thesis The Role of Usability in Software Validation – Case: Medical Device Manufacturing

Degree programme Degree Programme in Computer Science and Engineering

Thesis supervisor Prof. Marko Nieminen, Dr.Sc. (Tech) **Code of professorship** T-121

Department Department of Computer Science

Thesis advisor(s) Ninna Koivunen, M.Sc.

Date 1.10.2015 **Number of pages** 9+77+2 **Language** English

Abstract

This master's thesis follows a software project in a medical device company. The software is a manufacturing execution system (MES) which guides the production of medical devices and collects traceability data, including the device history files.

Because the software is used in the manufacturing of medical devices and in the implementation of the company quality system, the software needs to be validated for its intended use. The validation process must comply with the guidance and regulations of the Food and Drug Administration. The software system is implemented by a software supplier but the customer organization is responsible for the validation process.

In addition to fulfilling the regulatory requirements, the software is expected to increase productivity. In order to increase productivity, the software needs to be usable. This thesis uses previous research on medical device usability and on enterprise resource planning (ERP) system usability as guidance for improving the usability of the software system along with the human-centered design process and principles.

The target of the research was to find out how usability considerations can be integrated to a medical device software validation process; and how the medical device validation process contributes to the usability of the system. The research is implemented as a single case study, with the software project as the case.

The results suggest that managing use-related risks is an important part of the project. Sometimes regulatory requirements may over rule usability requirements, especially efficiency and flexibility. However, many of the principles of human-centered design turned out to be important success factors. Iterative process structure is a good choice. It is useful to have team members with different skills and backgrounds. End user involvement is needed throughout the project. Especially requirements gathering and design turned out to be critical activities for the project success. They are also activities which require input from end users.

Keywords Human-centered design, manufacturing execution system, medical device manufacturing, software validation, usability

Aalto-yliopisto
Perustieteiden
korkeakoulu

Aalto-yliopisto, PL 11000, 00076 AALTO
www.aalto.fi

Diplomityön tiivistelmä

Tekijä Emmi Kaivosoja

Työn nimi Käytettävyys ohjelmistovalidoinnissa – Tapaus: Terveydenhuollon laitteiden valmistus

Koulutusohjelma Tietotekniikan koulutusohjelma

Valvoja Prof. TkT Marko Nieminen **Professuurikoodi** T-121

Laitos Tietotekniikan laitos

Työn ohjaaja(t) DI Ninna Koivunen

Päivämäärä 1.10.2015 **Sivumäärä** 9+77+2 **Kieli** Englanti

Tiivistelmä

Tämä diplomityö seuraa erästä ohjelmistoprojektia terveyden huollon laitteita ja tarvikkeita valmistavassa yrityksessä. Ohjelmisto on tuotannonohjausjärjestelmä, joka ohjaa tuotannon työntekijöitä ja kerää jäljitettävyystietoja tuotannosta.

Koska ohjelmistoa tullaan käyttämään terveyden huollon laitteiden valmistuksessa ja laatujärjestelmän toteutuksessa, se tarvitsee validoida käyttötarkoitukseensa. Validointiprosessin tarvitsee noudattaa regulatiivisia vaatimuksia. Ohjelmiston toteuttaa ohjelmistotoimittaja, mutta validointivastuu on asiakasorganisaatiolla.

Regulatiivisten vaatimusten täyttämisen lisäksi ohjelmiston odotetaan parantavan tuottavuutta, mikä taas vaatii ohjelmistolta hyvää käytettävyyttä. Tässä diplomityössä hyödynnetään aikaisempaa tutkimusta terveyden huollon laitteiden ja toiminnanohjausjärjestelmien käytettävyydestä sekä käyttäjäkeskeisen suunnittelun prosessia ja periaatteita ohjenuorana ohjelmiston käytettävyyden arvioinnissa.

Tutkimuksen tavoitteena oli selvittää miten käytettävyysnäkökulma voidaan yhdistää terveyden huollon laitteiden ohjelmistovalidointiprosessiin. Lisäksi tavoitteena oli selvittää miten edellä mainittu validointiprosessi vaikuttaa järjestelmän käytettävyyteen. Tutkimus toteutettiin tapaustutkimuksena, jossa ohjelmistoprojekti on tutkimuskohteena.

Tulosten perusteella voidaan todeta, että käyttöön liittyvien riskien hallinta on tärkeä osa tällaista projektia. Toisinaan regulatiiviset vaatimukset yliajavat käytettävyysvaatimuksista, erityisesti tehokkuuteen ja joustavuuteen liittyvistä. Monet käyttäjäkeskeisen suunnittelun periaatteista osoittautuivat kuitenkin tärkeiksi menestystekijöiksi. Iteratiivinen prosessimalli on suositeltava valinta, samoin monialainen projektitiimi. Loppukäyttäjiä tarvitaan koko projektin ajan. Erityisesti vaatimusten määrittely ja suunnittelu osoittautuivat projektin onnistumisen kannalta kriittisiksi toiminnoiksi, joissa olisi tärkeää olla loppukäyttäjiä mukana.

Avainsanat käyttäjäkeskeinen suunnittelu, tuotannonohjausjärjestelmä, terveydenhuollon laitteiden valmistus, ohjelmistovalidointi, käytettävyys

Acknowledgements

Firstly I would like to thank my professor Marko Nieminen and my advisor Ninna Koivunen for providing me with their expertise and feedback during the thesis process. I would also like to say thank you to Jarno Yliluoma for all the practical thesis writing advice and motivational chocolates. A thank you goes also to all the kind people who gave up their time to answer to my questionnaires and interviews, and let me observe their work.

Finally, I want to say thank you to Ville and my other friends and family for keeping me sane and supporting me during the process. I could not have made it without you.

Espoo, October 1st, 2015

Emmi Kaivosoja

Table of Contents

Glossary

Enterprise resource planning (ERP) system	= A software system for managing the business of an organization
Human-centered design	= Iterative design which focuses on users and their needs
Manufacturing execution system (MES)	= A software system for controlling and guiding production work
Medical device	= An article, instrument, apparatus or machine that is used in the prevention, diagnosis or treatment of illness or disease, or for detecting, measuring, restoring, correcting or modifying the structure or function of the body for some health purpose. Typically, the purpose of a medical device is not achieved by pharmacological, immunological or metabolic means. (WHO)
Quality system / Quality management system	= A management system to direct and control an organization with regard to quality (ISO 9000)
Risk	= Combination of the probability of occurrence of harm and the severity of that harm (ISO 14971)
Usability	= Extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use (ISO 9241-11)
Usability engineering	= The process of producing usable products. Also used as a synonym for human-centered design.
User-centered design (UCD)	= See human-centered design

Validation	= Confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled (ISO 9000)
Verification	= Confirmation, through the provision of objective evidence, that specified requirements have been fulfilled (ISO 9000)

1 Introduction

Modern medicine uses all sorts of medical devices for treating and diagnosing patients. Medical devices range from more complex software devices, like patient monitors, to simpler thermometers and they are used in hospitals and homes alike. Patients and medical professionals depend on medical devices. A person's life may depend upon a medical device, so it is extremely important that the medical device is safe and works as intended. This is why medical devices around the world are regulated by laws and regulations. [1]

Since the quality and safety of medical devices is so important, there are also regulations for medical device manufacturing. A tool used in medical device manufacturing affects the quality of the final medical device; hence the quality of the tool is equally important. These days' software systems are often used in manufacturing. When a software system is used in medical device manufacturing, it needs to be validated for its purpose. [2, 3]

For a device or software to work in its intended use and be safe to use, it is not enough that it works technically correctly. If the device is so difficult to use that the person who should operate it cannot do it correctly, the device is not useful. In some cases incorrectly using a system can be dangerous. This leads us to the importance of usability. Usability is a safety factor especially for medical devices. [4, 5]

This master's thesis follows the manufacturing software project of one medical device company. The focus is on the effect that the validation process required for a software system used in medical device manufacturing has on the usability of the system; and how the validation process could improve the usability of the software system.

1.1 Background

The object of study for this master's thesis has been one software project in a medical device company. The software in question was a manufacturing execution system (MES) which is used to control and guide the manufacturing of medical devices. The system is also used for collecting and storing manufacturing history data which makes it part of the manufacturer's quality system. The system was developed by an external software supplier, but the customer was responsible for system validation according to quality system regulations.

The goal of the project was to improve the productivity by making the manufacturing work easier and to improve product quality by adding quality controls. In addition the software needed to keep accurate and complete records of manufacturing and comply with electronic records and electronic signatures regulations [6].

1.2 Research Questions

Although validation of medical device software, as well as software used in manufacturing of medical devices, must follow certain regulations, there is no rule that regulated software systems should not be usable. In fact, if the software systems are expected to improve productivity, the systems need to be sufficiently usable.

The target of this thesis is to examine how usability considerations can be taken into account in the software validation process for medical device manufacturing software. The secondary goal is to suggest concrete activities for software validation processes for medical device manufacturing, which improve the usability of the validated software system. Based on these goals, two main research questions with additional sub questions were formulated.

How does the validation process for medical device software affect the usability of the software?

The validation process for medical device software requires certain activities to be performed. These activities may or may not affect the usability of the software system. Some activities may improve the system usability while some may have an opposite effect.

> *Is usability in some situations over ruled by regulatory requirements?*
> Since the regulatory requirements for medical device software have a higher priority than usability requirements, it is examined whether or not these requirements contradict each other and if they do, in what kind of situations.

How can we produce usable, medical device regulated software?

This thesis tries to find ways to perform software validation for medical device regulated software that also address usability concerns for the software system.

> *How should the end users be involved in the validation process?*
> End user involvement is typically a corner stone for user centered design and usability methods. How do end users fit in the software validation process for medical device regulated software?

What kind of methods should be used, in which stage of the process and by whom?
Which are the actual methods that could be useful in the software validation of usable, medical device regulated software? When should they be used? Who should perform the activities?

What the software validation process should look like for medical device regulated, usable software?
What should the process model for usable, medical device regulated software be like? Are there some qualities that the process should or should not have?

Which are the critical points in the validation process?
Critical point in this question refers to the project stages, activities or attributes that affect the successfulness of the project. Successfulness is considered to be measured by the usability of the software system and the cost benefit ratio of the project. Regulatory requirements are not a factor in successfulness because fulfilling them is not optional but mandatory.

1.3 Research Implementation

The research for this thesis was implemented as a case study with one case, a medical device manufacturing software project, as the object of research. To support and complement the empirical data from the case study, background research data is also presented. Since there is not much research data about the usability of medical device manufacturing software or MES software, medical device usability research and enterprise resource planning (ERP) system usability research were used as the usability references.

1.4 Organization of the Thesis

The following two chapters present the background for medical device regulation and software validation, which are relevant for this thesis. Chapter two describes generally how and why medical devices are regulated presenting risk management and quality system regulation. Software and process validation, especially medical device related validation, is described in chapter three. Risk management, quality system regulation and software validation practices for medical devices are all vital components of the case study project.

Chapters four, five and six all discuss usability and user-centered design presenting the related research on usability topics which are similar to the case study project. Chapter four

gives a general summary of usability and user-centered design. Chapter five discusses usability aspects of medical devices and chapter six the usability of the ERP systems.

Chapter seven presents the case study research method, which is used as the research method in this thesis. Chapter eight explains how the case study was implemented and connects the background research to the case study project. Chapter nine presents the results from the case study project. Chapter ten includes the analysis of the results as well as the answers to the research questions.

2 Regulatory Requirements and Standards

Medical device industry is a regulated industry, which means there is a set of rules that a company needs to follow if they manufacture and/or sell medical devices. These rules are typically part of legislation or other form of governmental control. In United States, the medical device regulation is part of the code of federal regulations title 21, which covers food and drugs. The regulatory authority in charge of the area is Food and Drug Administration (FDA). In Europe, the medical device regulations can be found from directives 90/385/EEC [7], 93/42/EEC [8] and 98/79/EC [9].

2.1 Safety through Risk Management

Medical devices are commonly regulated to ensure the safety of the patients and other users. Regulation protects patients and other medical device users from malfunctioning and inefficient medical devices. Because of medical device regulation, people can safely assume that when they buy or use a medical device, the device will function in its intended use. All medical devices carry some level of risk and potential to cause problems under specific circumstances. Problems may occur due to device malfunction or inappropriate use of the device. In addition to safety, the performance of a medical device is an important factor which is often linked to the safety of the device. Let us consider a patient monitor as an example. If it shows false information to the medical stuff, the decisions based on false data can have devastating consequences to the patient. If the monitor shows correct data, but the monitor's alarm sound when something is wrong with the patient cannot be heard, the patient may again suffer severe consequences. Both of these examples describe performance failures which affect the safety of the medical device. [1]

Medical device manufacturers are required to analyze and evaluate risks associated with the use of the device. [2, 10] A risk consists of a hazard, which is a potential source of harm, and the probability and severity associated with the hazard, i.e. how likely the hazard is to happen and how bad are the consequences. The process of handling risks is called risk management. The manufacturers of medical devices have the responsibility to demonstrate to the regulatory authority that they have identified and addressed the possible risks associated with the use of the device. [1]

The degree of regulation imposed on a medical device depends usually on the potential hazards associated with it. Most of the regulatory systems classify medical devices in to three or four different categories which are subject to different levels of regulatory control. Attributes that can affect the classification of the medical device are e.g. the degree of invasiveness, duration of contact, the body system affected and local versus systemic effects. [1]

SFS-EN ISO 14971 [10] is an international standard, which describes how to apply risk management to medical devices. According to SFS-EN ISO 14971, risk management process consists of risk analysis, risk evaluation, risk control, and production and post production information.

Risk analysis and risk evaluation together form risk assessment activities. During risk analysis, the manufacturer is required to document the intended use of the device. In addition, the manufacturer needs to identify any reasonably foreseeable possibilities to misuse the device and the characteristics related to safety of the medical device. Based on the previously mentioned information, the manufacturer identifies and documents all known and foreseeable hazards associated with the use of the medical device. The hazards have to be considered both in normal conditions and in various fault conditions. All hazards are estimated regarding the probability and severity of the hazard, in case it would occur. [10]

Based on the estimates from risk analysis, the manufacturer needs to decide whether or not risk reduction measures, or risk control measures, are needed. The three possible approaches to risk control are, starting from the best, inherent safety by design, protective measures and information for safety. After the application of risk control measures the residual risk must be evaluated and if it is not acceptable, more risk control measures need to be applied. Also the risks arising from risk control measures need to be identified and evaluated in a similar manner. Sometimes, if the residual risk is not on an acceptable level but further risk control measures are not reasonable, the manufacturer can perform a risk-benefit analysis. If the medical benefits of the device outweigh the risk, the device may be acceptable for use despite the risk. [10]

Production and post production information refers to the manufacturer's duty to gather information about the device and its safety during and after production. This information is evaluated and if necessary, the risk assessment and risk control measures need to be reconsidered. [10]

2.2 Regulation as a Part of Medical Device Life Cycle

Medical devices are regulated in different ways throughout their life cycle, because any phase during their life span can affect the safety and performance of the medical device. The top row of figure 1 shows the main phases in life span of a medical device.

The first three phases: conception and development, manufacture and packaging and labeling are typically the manufacturer's responsibility. In the first phase, conception and development, the medical device is designed, developed, constructed and tested. Risk management is needed from the beginning to make a safe device. Thorough testing including verification, validation and clinical trials is also important. In manufacture phase the medical devices are manufactured. Manufacturing process needs to be controlled and managed in order for it to produce quality products consistently. Packaging and labeling can also affect the safety of the medical device. Inadequate packaging may result in a broken and malfunctioning medical device. Incorrect labels or use instructions can also compromise the safety of the patient.

Advertising and sale phases are the responsibility of the medical device vendor. Medical device advertising is usually regulated to prevent untrue claims about the performance of a medical device. Sale is the point when the medical device is put into actual use. If sale of medical device was not regulated in any way, anyone could sell medical device and the public would have no way of knowing which of the devices being sold are quality products and which are not.

Medical device users are usually responsible for the last two phases of the device life span: use and disposal. The use of the medical device has a great impact on the safety and performance of the device. Even if the device has no fault in it, a use error can cause a threat to the patients. Issues with device calibration and maintenance can also affect the safety and performance of the device. For some medical devices their correct disposal has a big impact on their safety. For example devices with toxic or contaminated materials must be disposed of correctly.

Life span phases	Conception and development	Manufacture	Packaging and labeling	Advertising	Sale	Use	Disposal
Manager of the phase	Manufacturer			Vendor		User	
Regulation stage	Pre-market			Placing on-market		Post-market surveillance / vigilance	

Figure 1: Medical device life span phases

The last row of figure 1 lists the different stages of medical device regulation and in which life cycle phases they are linked to. Different authorities have different names and systems for things, but the common idea behind them all is similar. This chapter describes the common framework for medical device regulation, not any specific system. In pre-market stage a medical device typically needs to fulfill the safety and performance requirements for the device, quality system requirements (see chapter 2.3) and labeling requirements.

Before a medical device can be put into market, most medical devices require a marketing clearance from the regulatory authority. The vendors who are in charge of the placing on-market stage are required to register with the regulatory authority. Vendors are also required to participate in the post-market surveillance. Medical device manufacturers are required to gather information about adverse events. Post-market surveillance studies can also be required either as a condition for product approval or after an adverse event. Manufacturers and in some regulatory systems also users are required to report any adverse events, that could result or have resulted in serious injury or death, to the regulatory authority. Post-market surveillance is related to the quality system requirements which are described in the following chapter. [1]

2.3 Quality System Regulation

All medical device manufacturers must fulfill some version of quality system requirements. In United States, the quality system regulation is part of the code of federal regulations [2]. In European Union, the requirements to have a quality management system can be found from council directives 90/385/EEC [7], 93/42/EEC [8] and 98/79/EC [9]. Quality systems are subject to auditing. In United States, the agency auditing medical device manufacturers is Food and Drug Administration (FDA). In European Union the audit is performed typically by an independent third party. [1]

World Health Organization's Medical device regulations – Global overview and guiding principles [1] defines quality system as "the organizational structure, responsibilities, procedures, processes and resources needed to implement quality management". ISO standards ISO 9000 and ISO 13485 [11, 12] use the term quality management system instead of quality system. ISO 9000 [11] defines quality management system as "management system to direct and control an organization with regard to quality". Quality system requirements can influence all phases during medical device life cycle: design, manufacture, packaging, labeling, storage, installation, servicing and post-market handling. The applicable requirements depend on the device and its risk level. [1]

The purpose of the quality system is to control the processes that affect the medical device during its life cycle. Carefully planned and controlled processes reduce the likelihood of non-conforming products and other risks related with the device. As opposed to reacting to issues found with the device while manufacturing or using it, the quality system is a preventative approach to medical device safety and quality. [1, 11, 12]

Quality system regulation requires documents and records to be maintained of practically everything that is related to the quality system compliance: device history, design history, complaints, procedures, reviews etc. These records are needed to prove to regulatory authorities that quality system regulations have been fulfilled. [2, 12]

Since software is becoming an integral part of modern medical devices, software used in medical devices affects the diagnosis or treatment of patients. Therefore the software itself and its development are scrutinized to ensure its safety and efficacy. [13] Software can be a medical device on itself or a part of a medical device. Quality system regulation applies to software in these cases just like it would apply to any other medical device. [2]

Quality system regulation specifically notes that design control section of the quality system regulation applies to devices automated with computer software. Design controls require that the manufacturer of the medical device establishes and maintains procedures for medical device design. These controls cover e.g. the process of forming requirements, design verification and design validation. The design process is documented in design history file, which is used to prove to authorities that procedures were followed during the design activities. [2]

FDA has published a separate document [3], which offers guidance to software validation to satisfy the validation, design control and general quality system requirements for software. General Principles of Software Validation [3] uses a wide interpretation of software validation activities. The document covers the whole software life cycle from initial requirements to software maintenance. FDA recommends integrating risk management into software life cycle management. This brings the safety aspect, which is critical for medical device and medical device manufacturing software, into software development process. The contents of General Principles for Software Validation [3] will be explained in more detail in chapter 3.3.

Quality system regulation states, that software which is used in medical device production or in implementation of the quality system must be validated for its intended use. [2] FDA's General Principles of Software Validation document applies also to software that is used in

medical device manufacturing or in implementation of the medical device manufacturer's quality system. [3]

When a software system is used to maintain records, e.g. device histories, the records in the software system are electronic records. Quality system records typically need to be signed. If a software system is used, that requires implementation of electronic signatures. Code of Federal Regulations title 21 part 11 [6] defines the requirements for electronic records and electronic signatures. These requirements must be met in order for electronic records and signatures to be considered equal to paper records with hand written signatures by the FDA. Computer systems for keeping electronic records are subject to FDA inspection.

3 Software and Process Validation

ISO 9000 [11] defines validation as “confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled”. Validation demonstrates that the object of validation meets user expectations, guidelines and other acceptance criteria. [14] Validation should ensure that the object of validation can be used for its intended purpose.

The object of validation can vary depending on the case. The quality system regulation [2], for example, mentions both process validation and device validation. FDA validation guidance, especially software validation guidance [3], has been criticized for confusing process and product validation. [15]

3.1 Process Validation

21 CFR section 820.75 [2] dictates when process validation is required. The rule is simple: If it is not possible to fully verify the results of a process, the process itself must be validated. Process validation activities and results need to be documented. Manufacturers are also required to have procedures to monitor and control the validated processes to ensure that the process continues to meet its requirements. Revalidation of a process needs to be considered if, for example, the process somehow changes or there are issues with the quality of the results produced be the process.

Before process validation can happen, a validation team is needed, preferably with multi-functional expertise areas. Process validation needs to be carefully planned and documented. Requirements for the validation need to be established as well as the approach to process validation. [16]

Process validation is divided into three phases: installation qualification (IQ), operational qualification (OQ) and performance qualification (PQ). Some sources [17, 18] add a fourth phase, which happens before IQ: design qualification (DQ). [16]

Design Qualification (DQ)
The purpose of design qualification phase is to make sure that the design is correct. It is important to verify early that the design corresponds to the requirements. Requirements consist of all requirements and guidelines that

concern the process, e.g. user requirements, validation plan, cGMP and applicable standards. [17]

Installation Qualification (IQ)
Installation qualification phase ensures that the installation is correct. The activities in IQ phase depend on the process and the process result. Calibration of process variables or equipment may be needed. If the process uses materials produced by a supplier, the supplier documentation may be inspected. If the process requires specific environmental conditions, such as temperature or humidity, these can be tested to be suitable for the process. The installation qualification phase should ensure that everything needed for testing the process is in place. [16]

Operational Qualification (OQ)
The purpose of operational qualification phase is to ensure that the process is guaranteed to produce an output that will meet its requirements. If the process has control limits, these need to be tested. Different kinds of failure modes and worst case scenarios should be experimented to see that the process will work correctly in these circumstances. User training can also be part of the OQ phase. After the OQ phase the process should be ready to be put into real use. [16]

Performance Qualification (PQ)
Performance qualification phase should demonstrate that the process continues to produce outputs that meet the requirements also in the long run. During PQ the acceptable variation in the process and the output should be established which will help with the process control. The PQ phase also tests the process repeatability. After the PQ phase the process should be stable and appropriate process controls are established for monitoring and maintaining the process. [16]

3.2 Software Validation

Software verification and validation (V&V) is performed in order to ensure the quality of the software. Verification ensures that the software or product meets the requirements for it. In other words, that the software works as specified. Validation on the other hand is concerned with whether the software meets the customer needs and requirements. In other words, does the software work in its intended use. Software testing is an essential part of V&V activities, but other activities, like reviews, are useful too. [11, 19]

V&V activities are performed during a software development process. There are a number of different kinds of software process models, but most of them include, with differing emphasis and detail level, the same abstract phases: specification, design, validation and evolution. Some software process models are more rigid, like the V-model, which is presented in Figure 2. In the V-model every step is completed before moving on to the next stage. Each level of V digs in to a more detailed level of the software. V&V activities are emphasized by creating V&V or test plan for every stage before the actual software development happens. [20]

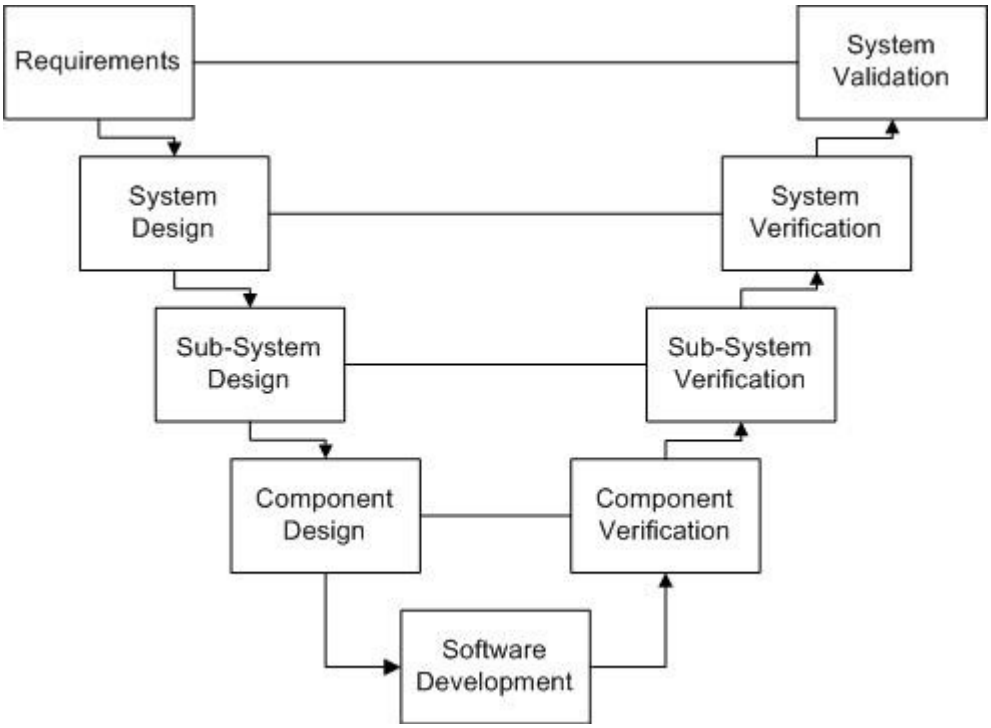


Figure 2: Software development process V-model [20, 21]

While the V-model has its perks, it has also disadvantages. There will be no prototypes or anything else concrete to show to customers until pretty late in the process, also there is no clear way to solve problems found during testing. Changes in project scope or requirements are also difficult to handle in a V-model. [20]

Agile software development models are a more flexible approach to software development. The idea of agile software development is to deliver working software frequently. Every delivery contains a little more features to the final product. The cycle of requirements, design, implementation and V&V are repeated for every software delivery. If the project scope or requirements change, the changes can be taken into account in the next delivery. Similarly any problems found during testing can be addressed in the next iteration. [19]

3.3 Software Validation Guidance for Medical Devices

The goal of software validation is to build a level of confidence that the software works in its intended use and does not contain defects that could cause unacceptable risks. This is by no means a trivial task. Software programs have the ability to branch to so many possible action sequences that it is not reasonable to test each and every possible sequence. Because complete testing of software is not a possibility, it is important to focus on perfecting the design and development process to produce quality software. Design and development are the critical phases to software quality when the majority of problems are introduced in to the product. [3]

Another critical phase is making changes to existing software. Software changes are also pretty easy and quick to do but they also carry a significant risk of inserting defects to the software. This is why software design and development processes should be tightly controlled and well planned. Change management as well as verification and validation after every change are important. [3]

These days not all software used in medical devices or in medical device manufacturing is specifically developed for that. Instead manufacturers may use already existing off-the-shelf software. In this case the manufacturer has no control over the software development process, but the manufacturer still has the regulatory responsibility. This means that the manufacturer of the medical device must ensure the safety of the off-the-shelf software, which can mean e.g. risk management activities, validation or auditing the software vendor. Off-the-shelf software needs to have documented requirements which define its intended use. [3, 22]

3.3.1 Principles of Software Validation

This chapter describes the 10 principles of software validation which are presented in the General Principles of Software Validation [3].

Requirements
Software needs to have documented requirements. Both software verification and software validation need requirements specifications to have requirements against which the verification or validation can be done.

Defect Prevention
Preventing the introduction of defects into the software during software development is important. Since it is impossible to thoroughly test software,

defect prevention requires planning and use of other methods, like reviews, in addition to testing.

Time and Effort

Software validation requires both time and effort. Therefore the planning and preparation for software validation should begin early on in the software life cycle. Evidence to conclude that the software is validated is gathered throughout the life cycle according to plans.

Software Life Cycle

General Principles of Software Validation does not recommend any specific software life cycle model, but one should be chosen and used. Software validation is tied to the software life cycle. Process validation framework presented in chapter 3.1 is sometimes used in software validation, but not as a rule.

Plans

Software validation is executed according to a plan. The plan defines and controls what the validation activities are meant to accomplish. Software validation plans include things like schedules, resources, scope, and validation activities.

Procedures

Defined procedures are used in the execution of the software validation. A procedure defines how a validation activity should be executed. Procedures define actions or action sequences used in software validation.

Software Validation after a Change

Every time a software change takes place the validation status of the software needs to be re-evaluated. Validation needs to be done not only to the changed feature but also to ensure that the change did not have any undesirable side effects. Regression testing is used to build confidence that the old features still function as intended after a change.

Validation Coverage

Validation coverage should depend on the risks associated with the software and the complexity of the software. High risk software needs more comprehensive validation than a low risk one. Validation efforts and their results should be found from the validation documentation.

Independence of Review

Independence of review dictates that quality assurance, like testing or reviews, should be conducted be someone who did not participate in creating the object of quality assurance activities. This is a good practice, because self-validation is difficult. It is easier to evaluate someone else's work.

Flexibility and Responsibility

Flexibility and responsibility means that the device manufacturer has flexibility in deciding how to apply the above mentioned principles. The implementation of the principles can differ between applications. Yet, the device manufacturer has the responsibility to demonstrate that the software has been validated according to these principles.

3.3.2 Software Validation Activities

Software validation consists of performing certain activities and tasks during different stages of software lifecycle. The selected software life cycle model dictates when and how many times these activities are executed. The software life cycle model should cover the software from the very beginning to its retirement. For very low risk applications some of the validation activities may not be necessary, but it is recommended that the following activities are at least considered when planning validation of software: [3]

- Quality Planning
- System Requirements Definition
- Detailed Software Requirements Definition
- Software Design Specification
- Construction or Coding
- Testing
- Installation
- Operation and Support
- Maintenance
- Retirement

4 Usability Engineering

Usability and focusing on users is sometimes seen as a luxury which can be left out of a development project. Yet, completely dismissing usability may lead to developing a useless product. In worst case scenarios, total neglect of users' needs and abilities may make it impossible to use the product at all. Users may also cease to use a product if it is awkward and difficult enough for them. [23]

Focusing on usability and users' needs has many possible benefits. Usable products can be more successful commercially than less usable counterparts. Usable systems can increase the productivity and efficiency of users and organizations. Systems with good usability require less user support. Sometimes usability focus is needed for safety, to protect users from risks to their health and safety. [24]

4.1 Usability – What is it?

Usability can and has been defined in multiple different ways. ISO 9241-11 [25] defines usability as "extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use." This short description has a number of noteworthy characteristics. First of all, the definition says "specified users" instead of just users. Making something work for everybody is pretty difficult. That is why the focus is on the target users of the product in question. "Specified goals" is a similar case. The product is not meant to be a universal tool, instead it needs to work well for the task it is intended for. Effectiveness is defined in ISO 9241-11 [25] as the "accuracy and completeness with which users achieve specified goals". In other words, a user must be able to perform the task correctly with the product. Since not just getting the job done is not enough, efficiency is part of the definition to add the expending of resources, e.g. time, to the mix.

Jakob Nielsen [26] depicts usability as a part of the overall model of system acceptability. Usefulness of the system consists of usability and utility. Usefulness on the other hand is part of the practical acceptability of the system, alongside with cost, compatibility, reliability and other attributes. System acceptability, on the top level, consists of practical and social acceptability.

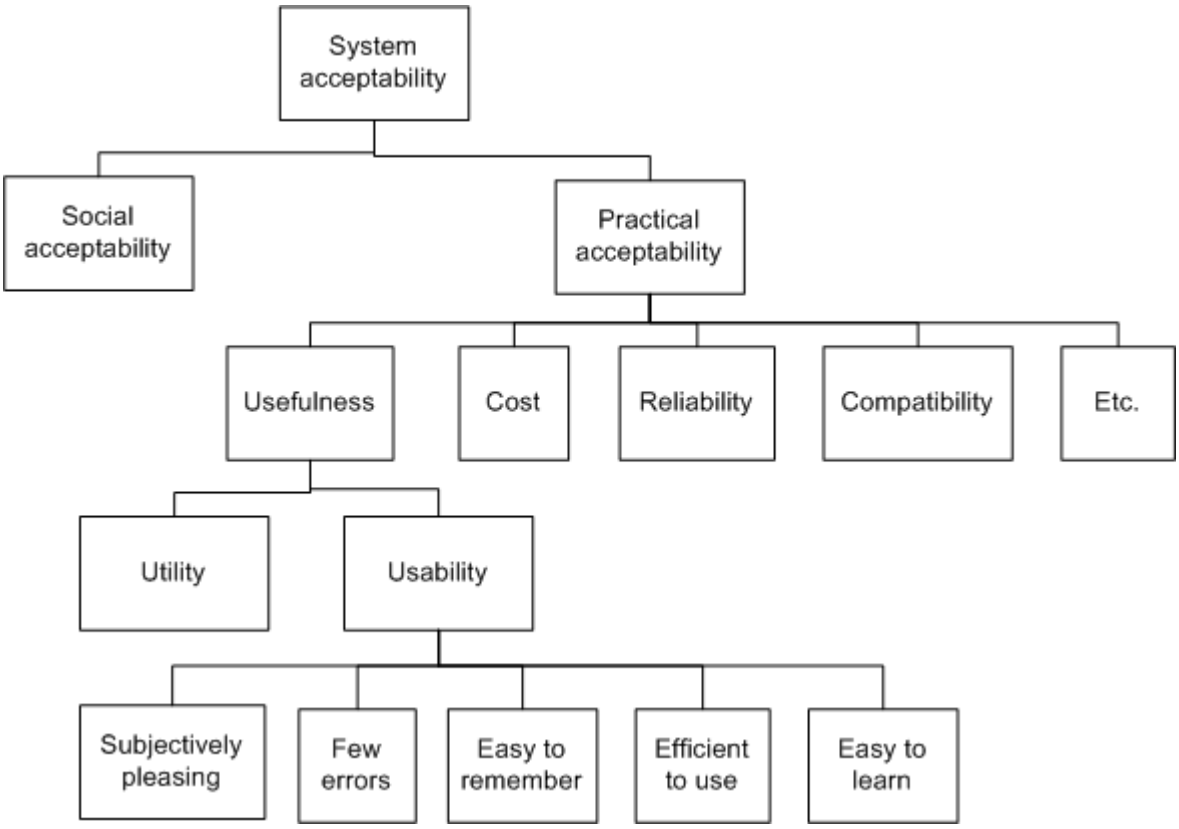


Figure 3: Usability as a part of system acceptability [26]

Nielsen defines usability as consisting of five attributes: learnability, efficiency, memorability, errors and satisfaction. Learnability describes how fast a new user is able to get something meaningful done using the system. Efficiency is related to the time it takes from an experienced user of the system to complete a task using the system. Memorability assesses how well a casual user of the system can remember how to complete a task using the system after some time has passed since they have done the task previously. Errors refer to the amount of errors users make using the system. The amount of use errors should be low and it should be easy for users to recover from them. Catastrophic errors should not occur. Satisfaction means the subjective opinion of the users, whether or not they consider the system to be pleasant to use. [26]

4.2 Usability Engineering

How do we make something usable? That is what usability engineering is for. Faulkner interprets usability engineering as “an approach to the development of software and systems which involves user participation from the outset and guarantees the efficacy of the product through the use of a usability specification and metrics”. Faulkner also describes usability engineering as “the entire process of producing usable products”. [23]

Usability engineering as a process that is used in product/system/software development is quite a common one. For example both Faulkner [23]; and Leventhal and Barnes [27] describe usability engineering as a process. The processes differ in detail, but the common phases are 1) information gathering about the users, their tasks and the context of use, 2) writing a usability specification based on the gathered information, 3) creating the design and 4) evaluating the design with users. The process is iterated as long as it takes to develop a design that is acceptable. A similar process model is also described in ISO 9241-210:2010 [24], although the standard does not use the term usability engineering but instead calls it human-centered design. In this thesis, term user-centered design (UCD) is also used to refer to the same process model.

In addition to the process steps, another common feature of the varying usability engineering process is the strong focus on the users and in understanding the reality of the users work. ISO 9241-210:2010 [24] describes the following as the principles of human-centered design:

> *a) The Design is Based upon an Explicit Understanding of Users, Tasks and Environments*
> In addition to recognizing the actual users of the system and finding out their characteristics, it is important to consider who else might be affected by the project outcome. They are stakeholders. Once again the context of use is an important factor. That might include the work practices, like frequent interruptions due to other work, or noise level.
>
> *b) Users are Involved throughout Design and Development*
> The users can not be involved in just some little part of the project, but be part of the project team from start to finish. Users need to actively participate, not just observe from the sidelines. The nature and frequency of their involvement can naturally vary depending on the project stage. Users can for example act as the experts sharing their knowledge about their work or actively take part in the design and evaluation activities.
>
> *c) The Design is Driven and Refined by User-centered Evaluation*
> The participation of actual end users is especially important in the evaluation of the design. The users are experts at their job, not anyone else. That is why they can better than anyone else evaluate whether or not the design works for them and comment on it. All feedback from users should be considered carefully.

d) The Process is Iterative

It is impossible to get a potentially complex system exactly right on the first attempt. That is why the process must be iterative. Each iteration is a possibility to get feedback from the users to improve the design. It is also possible that users will be able to express some of their concerns or requirements only when there is a design they can comment on.

e) The Design Addresses the Whole User Experience

This principle reminds us that the concept of usability is not just about making things not actively difficult for the user. The design should take in to account things like user's attitudes, personal goals, prior experiences etc. An example of what this could affect is deciding the extent to which something is automated in the system or performed by the user. It is noted that in safety or mission-critical systems effectiveness and efficiency may be considered to be more important than user satisfaction.

f) The Design Team Includes Multidisciplinary Skills and Perspectives

The design team should have a large variation of skills and perspectives to be able to see the design from different viewpoints. A multidisciplinary team can also better consider the effects of the project outside the obvious one that might not get noticed by a more homogeneous team. A multidisciplinary team is also more likely to develop revolutionary thinking outside-the-box solutions. Some examples of potentially useful expertise areas are: human factors and usability, users and other stakeholders, application domain/subject matter expertise, marketing, user support, user interface and product design, engineering.

The standard [24] defines a human-centered design process which consists of the following steps: a) understanding and specifying the context of use, b) specifying the user requirements, c) producing design solutions and d) evaluating the design. The steps are iterated until the solution meets the requirements. The process flow from one step to another is presented in figure 4.

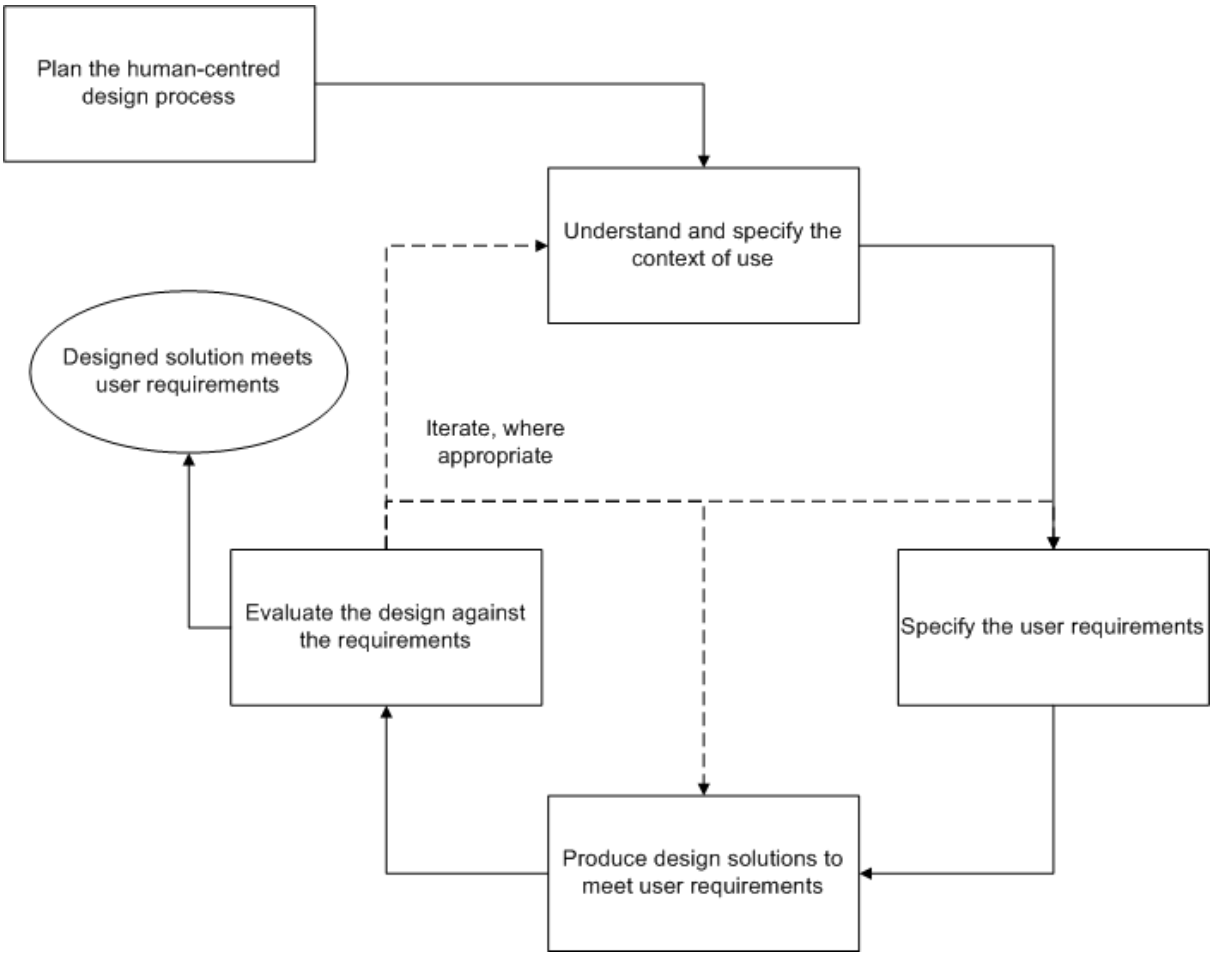


Figure 4: Process for human-centered design [24]

Understand and specify the context of use: The process begins with knowledge gathering. Who are the users? What are their skills, goals and work practices? Are there multiple different user groups? Where will they use the system and how? These questions along with many others help with understanding the context of use, which consists of the users with their various attributes, their tasks and the environment in which they work.

Specifying the user requirements: User requirements should take in to account user needs and context of use, as well as organizational requirements. Standards and guidelines may also be useful when specifying the requirements. In this stage, conflicting requirements should be identified and resolved. The result should be a requirements specification with measurable criteria for usability objectives in specific contexts of use.

Producing design solutions: In this phase, design solutions are produces based on the defined requirements. In early iterations the solution might be in form of a simple model or a paper prototype with which the design can be evaluated. In later iterations the design gets closer to the actual final solution.

Evaluating the design: In earlier iterations evaluation if important for getting feedback from users to guide the next iteration. The evaluation of proposed solutions with users may even bring up new user needs or requirements which they have not been able to express before. In the end evaluation is done to assess whether the solution meets user requirements. Even though evaluation with actual users is very important, expert evaluation methods, such as heuristic evaluation, can also be used.

4.3 Usability Engineering Methods

During a usability engineering process it is often necessary to gather information about the users, their tasks and the context of use. It is also useful to evaluate the possible design solutions. This chapter presents some methods used in usability engineering. This is not an exhaustive list, only the methods relevant for this thesis are presented.

4.3.1 Questionnaire

Questionnaire is one possible method for gathering information directly by asking from users. Questionnaire is a set of questions the user answers to. Questionnaires can be something user completes by himself or an interviewer might fill it while asking the questions from the user. Questions in a questionnaire can be open ended or have a limited number of possible answers. Questionnaires are useful for gathering subjective data, like users' opinions. They are not optimal for gathering objective data.

A common problem with questionnaires is the possibility that the questions are interpreted differently by each participant. That is why questionnaires require time for careful preparation. The questions should also be tested with a couple of sample users to see if there is ambiguity in the interpretation. A questionnaire, especially one with open questions, may produce a lot of diverse data which is difficult to analyze. The results of a questionnaire may become biased if the subject is embarrassing to the users or the questions make the users feel inadequate or incompetent. In these cases, the users may not answer entirely truthfully. [23]

4.3.2 Interviews

Interview, like questionnaire, is an information gathering method which asks users to answer questions. Interviews are usually done in person with the interviewer and interviewee in the same location. When comparing to questionnaire, interviews make it possible to rephrase questions in case there is reason to suspect that a misunderstanding has happened. This may help prevent bias from misinterpreted questions. Interviews share the issues with questionnaires regarding users not always being honest with their answers. Interviewees may

also give false information e.g. about what they do unintentionally, not being aware of every bit of data that matters.

Interviews can range from unstructured to structured ones. Unstructured interviews do not have a specific set of questions to be answered. They allow the interviewee to steer the interview and bring up topics that are important to them. Therefore unstructured interviews are well suited for explorative research where the interviewer does not yet know what they are looking for. In the other end of the spectrum structured interview has a prepared set of questions, and possibly also a prepared set on possible answers for the interviewee to choose from. Structured interview does not leave room for individual variation; instead it looks for gross response. Between the two extremes is semi-structured interview. It includes some prepared questions, but there is also room for additional questions and topics depending on the interviewees' answers. [23, 26]

4.3.3 Observation

Observation is method for gathering data about user's work in the realistic setting. It means actually going to see what the users are doing. Observation should intervene with the users' work as little as possible. The goal is to see the users' work as they would do it any other day. The observer should mostly just watch what is happening, although sometimes it is necessary to ask for an explanation to something the observer does not understand. Observation has a chance of giving more realistic picture of the users' work than their own accounts of it. It gives an opportunity to see unexpected ways of using a system that would have otherwise never occurred to the researcher. Yet there is a risk that being observed may, consciously or unconsciously, affect the users' actions. [23]

4.3.4 Contextual Inquiry

Contextual inquiry combines observation with unstructured interviews. For contextual inquiry, the researcher goes to the users' workplace to conduct one-on-one interviews with the users while they work. The goal is to thoroughly understand the details and motivations of users' work, by observation and by simultaneously asking the user to explain what they are doing and why. Contextual inquiry is originally part of contextual design, an approach to product design that emphasizes the understanding of customer's work. [28]

4.3.5 Heuristic Evaluation

Heuristic evaluation is a method for evaluating a user interface against a set of rules i.e. heuristics. The target of evaluation can be a finished product or a prototype. Heuristic evaluation is a form of expert evaluation i.e. evaluation without actual users. Heuristic

evaluation is called a discount usability method, because it does not require access to users and takes relatively little time and effort. [23, 26]

The evaluation has two phases. In the first phase, the evaluators will each individually go through the user interface or prototype against the heuristics. All violations of the heuristics are recorded during the individual evaluation. In the second phase the evaluators gather together to form a combined list of findings. Heuristic evaluation does not provide solutions to the recorded violations but since each violation is linked to the heuristic(s) it violated, it is usually possible to come up with a solution considering the heuristic. The heuristic evaluation should preferably be done by more than one evaluator. Evaluators tend to find different issues, which is why having three to five evaluators provides a better coverage than a single evaluator. [26]

There are a number of possible heuristics that can be used in a heuristic evaluation. There are Shneiderman's 8 Golden Rules and Norman's Seven Principles for Transforming Difficult Tasks into Simple Ones. The most well known set of heuristics are probably Nielsen's 10 heuristics, which are [23, 26]

1) Simple and natural dialogue
2) Speak the users' language
3) Minimize the user memory load
4) Consistency
5) Feedback
6) Clearly marked exits
7) Shortcuts
8) Good error messages
9) Prevent errors
10) Help and documentation

4.3.6 Usability Testing

Usability testing sometimes referred to as user testing, is a usability evaluation method. In a usability test, a user is asked to perform certain tasks using the system which is being evaluated. The user is observed during the test by the researchers. The user is also

encouraged to think aloud while they use the system to make it easier for the researchers to know why the user does something or if they are wondering about something. The tasks for usability testing are previously prepared and vary based on what is being investigated. Typically a similar usability test is run for more than one person to get more comprehensive data. Usability tests are typically done in a usability laboratory or other controlled environment, not at the user's work place. Usability tests can be used to collect both quantitative and qualitative data, e.g. identifying usability issues and tracking task completion times. The selection of users is important, since usability tests are usually performed with a very limited number of users. The selection of user participants for usability tests, e.g. novice or expert users, as well as whether the users should all be alike or have varied backgrounds and skill sets, depends on what is being studied. [26]

5 Medical Device Usability

The main reason to strive for usability for medical devices is the safety of patients and other device users. The goal is to minimize the likelihood of use errors and use-related hazards, which in worst cases can lead to fatal consequences. Use error means using the device in a way that leads to unintended results. [29] Hazards caused by use errors are called use-related hazards. [5] Use errors are becoming a significant factor that affects medical device safety. It is suggested that frequency and consequences of use-related hazards might far exceed the hazards arising from device failures. [5]

With the growing awareness of usability related issues, the authorities are also beginning to require that the manufacturers take action to prevent the issues. FDA has replaced the term user error with use error, to emphasize that the users are not to blame when accidents happen because of poor usability. [30] This gives the manufacturers the signal that it is no longer enough to consider risks related to device failures. Also use-related failures need to be addressed. [4] The manufacturers are also required to submit evidence of including use errors and their control measures in risk management activities. [30, 31] The design control requirements in quality system regulation mentions addressing user needs and the intended use of the device during design input and design validation. [2, 30, 31]

Although there are requirements for improving medical device usability and safety, more awareness of the issues is needed in the industry. Acharya at al. [4] claim that there are not enough people with human computer interaction (HCI) skills in the medical device industry. This leads to medical devices with use-related risks which could have been found and corrected with basic HCI practices. Viitanen and Nieminen [32] point out the importance of planning ahead and considering the big picture from the user's point of view in medical device software projects. They claim that the focus of usability related activities is too much on evaluation, when it should be on planning and design phases of medical device software projects. If issues are found when the device or software is almost ready, mending them is expensive.

One major obstacle for applying usability engineering in medical device industry is the lack of support from upper management. Usability engineering activities require time, access to users and support from other disciplines inside the company. This is why the management should be aware of benefits of usability engineering and advocate for it. [30, 33]

5.1 Usability Engineering and Risk Management

5.1.1 Usability Engineering Process for Medical Devices

There are at least two standards which describe a usability engineering process for medical devices. There is IEC 62366 Medical devices – Application of usability engineering to medical devices [29] and IEC 60601-1-6 Medical electrical equipment – General requirements for basic safety and essential performance – Collateral standard: Usability [34]. The standards describe a recommended usability engineering process for medical devices which is meant to ensure that the remaining residual risk is at an acceptable level. The process models in these two standards resemble each other. This chapter presents the process according to IEC 62366.

The standard requires the manufacturer to establish, document and maintain a usability engineering process throughout the entire life cycle of the device. The documentation of the usability engineering process forms the usability engineering file, which is also a mandatory requirement according to the standard. [29]

The usability engineering process in "IEC 62366 Medical devices – Application of usability engineering to medical devices" is described as having the following steps:

1) Application specification
2) Frequently used functions
3) Identification of hazards and hazardous situations related to usability
4) Primary operating functions
5) Usability specification
6) Usability validation plan
7) User interface design and implementation
8) Usability verification
9) Usability validation

Steps 1-4 are mostly preparation for forming the usability specification (step 5). The application specification is meant to answer questions like who, why, when and where is going to use the device. Step 2, "Frequently used functions", requires the determination of the most often used functions of the medical device that involve user interaction with the

medical device. Step 3 is about performing a risk analysis for the device, focusing on usability related risks and possible use errors. During step 3 the characteristics of the device that are related to its safety are identified. The primary operating functions (step 4) are determined based on frequently used functions (step 2) and characteristics related to safety (identified in step 3). In the usability engineering process the results of the previous steps are often used to execute the following steps in the process. [29]

In step 5 the usability specification is created. Usability specification includes testable requirements for the primary operating functions and also the criteria for evaluating the adequacy of risk control measures. The usability specification must include at least use scenarios related to primary operating functions, user interface requirements for primary operating functions and requirements for determining if the user can easily recognize the primary operating functions. [29]

Based on usability specification a usability validation plan is created in step 6. The usability validation plan presents methods and criteria used for validation of the usability of the primary operating functions. The plan also describes the planned involvement of representative users in the validation. The standard specifies that both quantitative and qualitative validation methods may be used. It is also noted that usability validation may be performed in laboratory setting, in a simulated use environment or in an actual use environment. [29]

In part 7, user interface design and implementation, the user interface is implemented as described in usability specification. The created user interface is evaluated during steps 8 and 9, usability verification and validation. Verification means evaluating the user interface based on the usability specification to make sure that the user interface meets all the requirements in the specification. Usability validation is executed according to the usability validation plan. The accompanying document (for example instructions for use) is subject to validation as well as the user interface. If the validation criteria are not met, the options are either to improve user interface design (return to step 7) or assess whether the benefits of the intended use of the medical device outweigh risk arising from usability problems. The latter consideration may lead to the device being acceptable if the risks are estimated to be at an acceptable level. [29]

The usability engineering process described in IEC 62366 [29] is meant to be used side by side with the risk management decision making process described in ISO 14971 [10]. Figure in appendix A describes the interaction between the processes.

5.1.2 Use-related Risks

Kaye and Crowley [5] claim that medical device designers do not consider use-related risks enough. Instead, the focus in risk analysis is on device and component failures. It is suggested that human factors engineering approach should be integrated into risk management process. This would make it easier to identify, control and prevent use-related risks. Use-related risks usually occur because something not anticipated happens in the use situation of the device. The unanticipated aspect can be many things, like user's actions, user's capabilities or use environment. The goal of incorporating human factors engineering into risk management is to be able to anticipate the previously unanticipated risks in device use.

Three factors affect the safety and effectiveness of the medical device use. These are use environments, users and the device itself. All of these need to be considered in the management of use-related risks. The use environment can affect the use situations with e.g. distractions or adding to the mental or physical workload of the user. Users have different kinds of abilities, expectations and limitations which all can affect the device use. The medical device can affect the safety and effectiveness of the use with e.g. complexity of the system and safety mechanisms which make recovery from errors easier.

Kaye and Crowley [5] present a process for addressing use-related hazards in risk management. The process is depicted in figure 5. The process begins with identifying scenarios that can lead to use-related hazards. Then the risks associated with the hazards are assessed and prioritized. Mitigation and control strategies are developed and implemented for risks if they are needed. After that the mitigation and control measures are verified. If the risks from use-related hazards are not acceptable after that, more mitigation and control strategies are needed. If the risk after verification of mitigation on control strategies is acceptable, the introduction of new use-related hazards needs to be evaluated. If new use-related hazards have been introduced during the process, the process needs to start from the beginning. If there are no new use-related hazards, it is time to validate safe and effective use of the medical device.

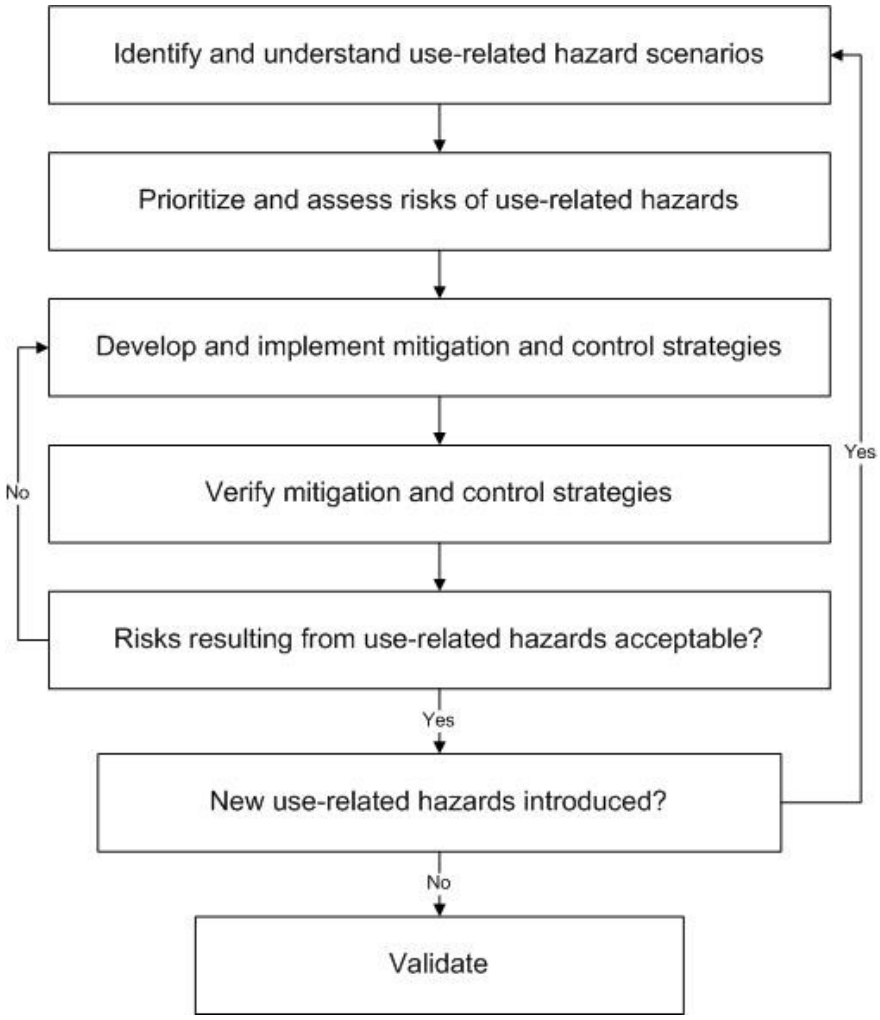


Figure 5: Management of use-related risks

Just like ISO 62366 [29] includes descriptions of device use, users and use environments, Kaye and Crowley [5] also recommend a device use description as a tool for identifying use-related hazards. In addition analytic and empirical approaches are recommended to help identify more use-related hazards. Analytic approaches involve using systematic methods, like function or task analysis, for analyzing the device use. Empirical approaches consist of methods which include actual or simulated use of the device, like usability testing or walk-throughs. Analytic approaches can identify the so called anticipated hazards with the device use. Empirical approaches can find hazards which could not be found with analytic approaches. Empirical approaches can also be used to get more information about an analytically identified hazard or to test the adequacy of mitigations. Analytic approaches, on the other hand, can find rarely occurring hazards which would not manifest themselves during empirical approaches. Analytic and empirical approaches complement each other and are recommended to be used together. [5]

5.1.3 Alternative Approaches

IEC 62366 [29] has been criticized for being too abstract and unclear. [33, 35] The usability engineering process descriptions in the standard include inconsistencies, which make it more difficult to apply. The interaction between usability engineering and risk management processes is claimed to be insufficient. The standard lacks qualifications for the usability engineer and does not give instructions or requirements for developing a mature and functional usability engineering process. [33]

Freudenthal et al. [33] have suggested improvements to the IEC 62366 standard [29] regarding the division of responsibilities and the process model. They have concluded that usability engineering is not easy. Therefore the usability engineering process and methods should be managed by a trained usability engineer. They have also come to a conclusion that the management of use-related risks should be the responsibility of the usability engineer instead of the risk manager. This reason for this is that the management of use-related risks requires using usability engineering methods, with which the usability engineer is already familiar with. It is expected that it is easier for the usability engineer to learn to focus on risk analysis and control than it is for the risk manager to learn the methods and mentality of usability engineering.

For the usability engineering process for medical devices, Freudenthal et al. [33] suggest that the standard should make it clear that the usability engineering process iteration can be used both as an overarching development process, which is iterated, and as a sub-process, which is iterated in various phases of the development. They have demonstrated the successful use of the latter version in a strictly linear product development project. It is also suggested that the usability engineering process needs more feedback loops to earlier phases. The standard's process model may give the impression that requirements cannot change once they have been created. This should not be the case. New information gathered during one iteration should be able to affect the requirements for the next iteration.

Many sources [5, 32, 35] emphasize the importance of early consideration of usability and safety. Evaluating risks or usability issues when the product is almost ready is much less efficient than considering these things during the requirements and design. If an issue is found late, fixing cost is usually high. If there is no more room in the budget, the fix may be just an added warning, or nothing. If the same issue is considered earlier, the product may originally be designed to prevent the issue from occurring.

Thimbleby [36] has claimed that usability engineering methods, while useful, are not sufficient for safety critical systems. Usability engineering methods cannot handle the

complexity of the devices, which can have hundreds or thousands of different states. As a solution Thimbleby suggests using formal methods e.g. for the analysis of device states and transitions between states. Formal methods can find issues with state transitions more reliably and with less effort than usability engineering methods.

5.2 Methods of Medical Device Usability Engineering

Usability engineering for medical devices uses many of the same methods as the traditional usability engineering, either as is or with some modifications. The methods range from drawing information from users; like contextual inquiry, interviews, questionnaires and usability testing; and expert evaluations, like heuristic evaluation and cognitive walkthrough; to analysis methods, like task analysis and use error analysis. No single method is a silver bullet which suits every situation. Usually several methods need to be used together to get the best results. The right methods are likely to be slightly different for each case. Because the records may be audited, methods that generate objective data are preferred. When methods which require research participants are used, it is of vital importance to have representative participants who perform the tasks under evaluation. [29]

The next two chapters present in more detail two methods suggested for improving medical device usability: heuristic analysis and Appraisal and Measurement of User Satisfaction (AMUSE). Both of these methods are based on quality attributes which are deemed to be desirable in medical devices.

5.2.1 Heuristic Evaluation

Zhang et al. [37] have proposed heuristic evaluation as a method for evaluating safety features of medical devices. Heuristic evaluation as a general method has been described in chapter 4.3.5. The proposed heuristics for medical devices are a combination of Nielsen's and Shneiderman's heuristics. The heuristics are

1) Consistency and standards
2) Visibility of system state
3) Match between system and world
4) Minimalist
5) Minimize memory load
6) Informative feedback

7) Flexibility and efficiency
8) Good error messages
9) Prevent errors
10) Clear closure
11) Reversible actions
12) Use user's language
13) Users in control
14) Help and documentation

Heuristic evaluation is recommended as a method which is easy to use and learn. It does not take a lot of time and it can find a great proportion of larger usability problems. The limitations of the method are that it does not reveal missing functionality or indicate which elements of the system under evaluation follow usability guidelines. Heuristic evaluation also does not find issues related to use environment. It is recommended to combine heuristic evaluation with other methods, like observation or usability testing to find most of the major usability issues. [37]

5.2.2 AMUSE

Doerr et. al [35] have suggested using the Appraisal and Measurement of User Satisfaction (AMUSE) methodology in medical device requirements engineering. The goal is to factor in usability and user-perceived product quality early on in the development process. In the heart of AMUSE methodology is a quality model, which is based on quality in use -metrics in ISO/IEC 9126-4 [38] and technology acceptance model (TAM) [39]. [35, 40] In AMUSE quality model the overall user satisfaction consists of five quality aspects: effectiveness, productivity, trust, hedonic quality and manufacturer's service. [35] The first four of these are closely related to the quality in use metrics in ISO/IEC 9126-4 [38]: effectiveness, productivity, safety and satisfaction.

- *Effectiveness* represents the ability to achieve intended results with completeness and correctness. Completeness means that it is not necessary to perform additional manual tasks even in a worst-case scenario. Correctness means the absence of errors when operating the system. The system should be robust enough to allow easy recovery from various possible errors. [35]

- *Productivity* refers to effort needed to successfully complete a task using the system. [35]

- If users *trust* the system, they consider the risk of damage to themselves, to other people and to business to be low when they use the system. [35]

- *Hedonic quality* describes the system's ability to satisfy the human need for stimulation and identification. For example it should be possible to experimentally explore the system without risk of causing damage. [35]

- *Manufacturer's service* is part of the system package to the users. If users perceive the manufacturer having put a lot of work in to fulfilling users' expectations, they are more satisfied with the system.

The AMUSE method is designed to aid in the selection of features for a product. AMUSE questionnaire can be used to measure the current state of each quality aspect in a product. This will give an idea about which quality aspects need improvements. Potential new features are rated based on how they will affect each of the quality aspects. This is called appraisal. Appraisal outcome can then be used in feature prioritization. [35] AMUSE is best suited for enhancing existing products. It can also be used to compare two or more products. AMUSE is more difficult to use for a new product, because in that case there is no base score for quality aspects from which to work on. [40]

6 Human Factors in ERP System Implementation

Enterprise resource planning (ERP) systems are business management software used to manage business data of a company. Typical data handled in the ERP system are e.g. inventory, orders and purchase orders, and manufacturing data. The reason for discussing ERP systems in this thesis is their similarity with MES systems, or at least the MES system project studied in this thesis. There is not much research yet about MES systems, especially about MES usability, but ERP systems and their usability has been studied previously. Both of these systems support important processes in the company and can have a large effect on the processes. It is expected that system implementation will increase efficiency and productivity of the company regarding the processes which are handled by the new system. Both ERP and MES systems are used to integrate scattered information and actions in one central system. This is expected to improve the availability of information. The project models are also similar: the system is purchased from an external vendor and customized for the customer.

Although ERP systems are recognized as valuable tools for business, they are not considered to be easy to use. Many sources comment on the complexity of ERP user interfaces which makes it difficult to learn to use them. [41, 42, 43] ERP systems have also been criticized for making information finding tedious [42] and not providing enough guidance to users [41, 42, 44]. In addition to ERP systems being not user oriented enough, human factors issues rise also from ERP implementation project practices. ERP system implementation often means big changes for the users. Failing to provide the users with sufficient training or introducing the new work patterns to users only when the system is deployed does not improve the user acceptance of the new ERP system [41, 45]

ERP usability and human factors aspects have not been a widely researched topic, probably because ERP systems have been created for business processes and not to please users. [43] However, ERP systems and ERP projects could benefit from human factors approach, because user errors in ERP use can have far-reaching consequences to business. [43] There is also the risk that if the ERP system is difficult and unappealing enough to users, they might not use it but instead find alternative ways to do things. [46]

Minimizing these aforementioned problems is not an easy task. Previous studies suggest some possible methods and tools for this. Some studies focus on ERP system characteristics that affect system usability. Other studies concentrate on methods used for evaluating ERP usability or on implementations of ERP projects from the user centered perspective.

6.1 Usability Engineering Process in ERP System Implementation

Vilpola [47] has suggested integrating user-centered design (UCD) process in to ERP system implementation project. She claims that since the goals of the user-centered design are similar to the goals of an ERP implementation project, the principles of user-centered design are applicable to ERP implementations. One of the main goals for an ERP project is typically improved efficiency, which is also a significant factor in user-centered design.

Vilpola [47] has integrated user-centered design process in to an ERP implementation process description by Mäkipää [48]. The ERP implementation process has 8 + 2 stages as described in figure 6. Initiative stage is the start of the ERP implementation project, where the need for an ERP system is recognized. In evaluation stage the processes and requirements for the ERP system are gathered. This contains both requirements gathering inside the company and also gathering information about the available ERP system vendors. After that the vendor and ERP system are selected. [47]

Between the selection and go-live, when the use of the system begins, there are several tasks that need to be performed. Existing processes of the company and the ERP system have to fit together. This requires modifications to the system, business process reengineering (BPR) or both. Modification may consist of code level tailoring or setting up parameters and configuration. Modification may also include the building of interfaces with existing systems. The existing data needs to be transferred to the ERP system. The intended users of the ERP system need training in the use of the ERP system. [47]

The go-live phase can happen in a big bang or the system can be taken in to use incrementally or in phases. After the go-live, the project ends, or is supposed to end, with a termination phase. However, there is evidence that the project rarely is considered fully completed. Instead, the implementation of the system functionality goes on as a continuing process. The roles people have had during implementation or BPR tend to stick even though their main responsibilities are elsewhere. After the implementation project is finished, it is recommended that the companies keep working on improving their utilization of the system to make the most of it. [47]

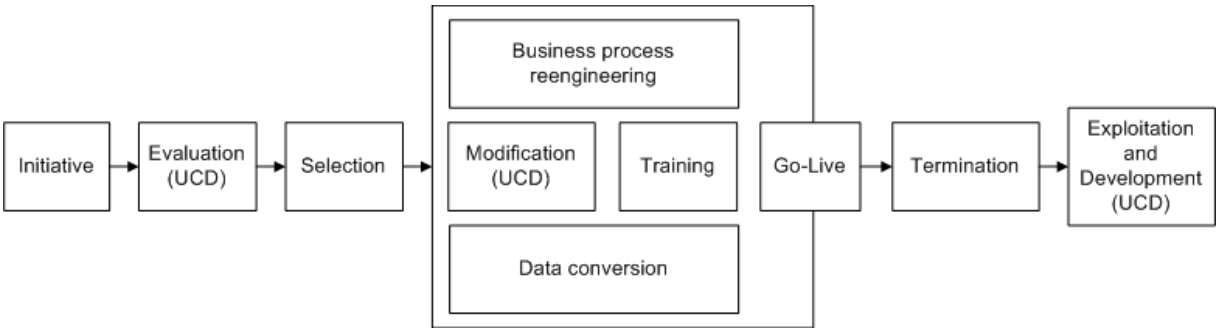


Figure 6: UCD in ERP system projects [47]

Vilpola suggests applying user-centered design process, which is described in chapter 4.2, in three phases of an ERP implementation project: evaluation, modification, and exploitation and development. [47]

In evaluation phase the UCD process is used to help choose the correct ERP system. ERP systems are often commercial-off-the shelf (COTS) software products, which means they have not been built to the exact requirements of the customer organization. This makes the choice of the most suitable system important. The UCD in evaluation phase helps in understanding and defining the context of use both in current state and in planned future state. This information is turned in to user and organization requirements. The requirements are the basis for designing target work processes and illustrating how the ERP system should support the processes. The designed work processes are evaluated against the requirements and the functionalities of the potential ERP system. This indicates whether the potential ERP system will be able to support the planned processes in the company. [47]

The UCD process in modification phase covers both modifications to the system and changes to existing business processes, i.e. BPR. The goal is to fit the selected ERP system functionalities and the internal processes together in the best possible way. Modifications to the system are error prone and can cause problems with project schedule and future software updates. This is why the potential of BPR should not be overlooked. The UCD process in modification phase starts when a need for change is recognized either in ERP system, existing processes or both of them. Again, the context of use for the business process or ERP system functionality needs to be understood. The next step is specifying the user, organizational and ERP system requirements. Here especially the end users should be closely involved. Based on the requirements a system modification and/or a BPR solution is designed. The design is evaluated against requirements. It is strongly recommended to include end users also in the design and especially evaluation stages for two reasons. The first reason is to get early feedback to avoid problems during deployment. The second reason is to see user participation as a form of change management. Users are more likely to accept the new

system and changes if they are active participants in the change, instead of being forced into it. [47]

The UCD process in the exploitation and development phase is similar to the process in modification phase, except that modifications to the software are mostly off the table. Instead, the UCD process is used to further improve the work processes and increase efficient utilization of the ERP system. [47]

6.2 Other ERP Usability Methods

Several previous studies have experimented with different kinds of methods for evaluating ERP usability in order to improve it. Scholtz, Cilliers and Calitz [41] recommend combining several different methods. Used to together, complementary methods can give a more complete understanding about the system than any one method. This is called triangulation. One could use for example a questionnaire with a large number of participants and then get a more detailed view about an issue rising from the answers with in depth interviews with a smaller number of participants.

Topi, Lucas and Babaian [44] suggest that ERP usability will improve if collaboration is used as the model for the system design. The idea is that the user and the ERP system are collaborating with each other to achieve a goal, i.e. perform a task. The three principles of collaborative behavior are commitment to mutual support, commitment to joint activity and mutual responsiveness. In their study they were able to connect common ERP usability issue categories to violations of one or more of these three principles.

Inka Vilpola and Kaisa Väänänen-Vainio-Mattila [45] suggest a thorough contextual analysis at the start of the ERP implementation project to get a good picture of the organizational aspects. Contextual design has also been suggested as a tool to help choose the right ERP system for the use context. [49] Vilpola and Väänänen-Vainio-Mattila [45] have also suggested including end-users early in the project so that their input can be considered early enough during the project.

6.3 ERP Usability Characteristics

Parks [43] has studied the effect of ERP user interface complexity on system usability. In this case, the usability attributes evaluated are the user's ability to successfully complete a task and the time it took to achieve it. These are expected to be the most important usability factors for an ERP system. Users completed tasks using both an existing ERP software interface and a specially built simplified interface. The complexity of the two user interfaces

was evaluated using two different methods, which both indicated that the default interface was much more complex. According to this research, screen complexity does not significantly affect the user's ability to complete a task. It does, however, affect the time it takes to complete a task. Users completed the task faster using the simpler user interface than the more complex one. The simpler interface, which also included more guidance, got positive feedback from the users. The default interface received comments about leaving the user feeling puzzled what to do next. As an unexpected result, the researchers noticed that all test participants, who succeeded in their task, checked their data before submitting.

Calisir & Calisir [46] have studied which user interface usability characteristics contribute to user satisfaction. Based on their research, perceived usefulness is the biggest contributor to user satisfaction. To a lesser degree, learnability was also a notable factor. Based on this study, perceived usefulness is affected by perceived ease of use, system capability and user guidance. User guidance was also a factor which affected the learnability of the system. Calisir & Calisir also suggest multiple different interfaces for ERP systems to cater to the needs of a diverse user population.

Singh and Wesson [42] have a slightly different approach to the subject. They have proposed a set of heuristics and evaluation criteria for ERP systems. Their study shows that traditional heuristics (like Nielsen's 10 heuristics) do not reveal the whole truth about ERP system usability issues. ERP heuristics brought to light different problems which were significant to the users. Using both their ERP heuristics and Nielsen's heuristics together gave a good overall picture about the system's usability issues. The Singh and Wesson [42] ERP usability criteria and corresponding heuristics are:

Navigation

The corresponding heuristic is Navigation and Access to Information. This heuristic is used to determine whether the user can find and correctly identify information and functionality in the system. Another aspect is the system's ability to guide the user to complete a task. Guidance information and search functionality are also worth considering.

Presentation

The corresponding heuristic is Presentation of Screen and Output. This heuristic is used to evaluate the layout, menus, dialog boxes, controls and other information presented on the screen. The information presented on screen and in reports should be easy to understand and interpret.

Task Support

The corresponding heuristic is Appropriateness of Task Support. This heuristic takes in to account the alignment between the system and the real world. System terminology and user terminology should be consistent with each other. The system should automate routine tasks and improve user productivity. The system should provide real-time information fast enough.

Learnability

The corresponding heuristic is Intuitive Nature of System. This heuristic describes how easy it is for a new user to learn to use the system efficiently. The system should have a sufficient on-line help. The user should be able to explore the system and identify the function or information they are looking for.

Customization

The corresponding heuristic is Ability to customize. Customization of the system enables the system to match the business processes of the users more accurately. Business processes are bound to change over time and customization makes it possible to change the system accordingly. This heuristic covers also the ability to customize the system on a single user level, to better match the needs of different types of users.

7 Case Study Research Method

A case study is a qualitative research method, which focuses on one or more subjects, i.e. the cases, which are examined in detail. The case study is a research strategy which originates from social sciences but can be used for many different kinds of research areas. As opposed to an experiment, which typically is conducted under controlled circumstances, a case study investigates a phenomenon in its real life context. Therefore the case study method is especially suitable for phenomena which cannot be separated from their context. In a case study there are likely to be more points of interest than there are data points. Because of this, case study relies on multiple sources of data, which can include qualitative and/or quantitative data sources. The purpose of multiple data sources is data triangulation, which means that more than one source of data supports the same finding. [50]

Because a case study investigates one or few subjects in detail, but the case study is suitable for research which focuses on finding out reasons or focuses on methods. In other words, a case study is recommended when the research questions are “why” or “how” questions. Another research characteristic that advocates the use of case study method is focus on current events. Since case study examines a phenomenon in its context instead of controlled circumstances, the case study is suitable when the investigator has little or no control over the phenomenon under investigation. [50] Bensabat claims that yet another characteristic that separates case studies from alternative methods, e.g. field experiments, is that a case study researcher may have less previous knowledge about what the variables of interest will be and how they will be measured [51].

Yin [50] mentions three types of case studies: exploratory, descriptive and explanatory. Exploratory case study focuses on a phenomenon with little or no previous research knowledge. It can be used e.g. to test a theory. Descriptive case study focuses on creating detailed descriptions about the case(s) being investigated. This can be used e.g. to describe a phenomenon in its context. Explanatory case studies focus on causal relationship and finding an explanation to a phenomenon. A case study typically uses prior propositions or theories which guide the collection of and analysis of data. An exploratory case study might not always have a prior theory but some initial ideas are necessary to focus the data collection activities and to determine whether the study was successful or not. [50] Case studies can also be used to build theories based on the research. [52, 53]

The case study research method has not been used very much in information system research. Still, case study research method is well-suited for information system research especially when the focus of the research is organizational rather than technical. [51]

7.1 Planning a Case Study

There are a number of things to be considered in designing a case study. The first issue is deciding on the study's questions. These are the questions that the case study is looking to answer. They define what the study is about. The second important task is thinking about the propositions related to the research questions. Propositions, or theories, are needed to guide the data collection. A question alone might not give any indication what kind of data should be collected, but a proposition gives ideas what kind of data could support or rebut the initial proposition. [50]

The third big decision in case study planning is the unit or units of analysis. This defines what will be the subject of investigation: company, project, group of people or something completely different. When choosing the unit of analysis and the specific case(s) to be studied, it is especially important to remember the research questions. The unit(s) of analysis and specific case(s) to be investigated has to fit the research questions. [50] The cases in case studies are not random samples; they are selected because of their suitability. [50, 52]

Some less evident but still important considerations in case study design are logic linking data to the propositions and the criteria for interpreting the findings. There are no established practices for how these should be done, which makes these issues a bit more difficult. One possible way to link data to propositions is patter-matching, which is described in chapter 7.3.2. Some other method for finding links between theory and data can be used as well. Criteria for interpreting the findings refer to criteria that define whether the data fits the theory well enough. A statistical test is often not an option for case study data. One way to address the issues is a thorough consideration of rival theories. If the data fits the chosen theory better than any of the rival theories, that makes the results more convincing. [50]

7.1.1 Case Study Designs

A case study research can have one or more specific cases which are investigated. A single case study design has only one case; multiple case study design has two or more cases. A case can contain one or several units of analysis. A case which contains more than one units of analysis is called an embedded case. A case with only one unit of analysis is called a holistic case. An embedded case study design can be used e.g. if the case is a public program which consists of projects. In this case, the projects would be the embedded units of analysis. Both

embedded and holistic case study designs can be used with single and multiple case study designs. [50]

Single and multiple case designs, as well as holistic and embedded case study designs, all have their strengths and weaknesses. A multiple case study design is often considered to be more convincing because there is a wider set of data and everything does not depend on one case giving all the answers. Naturally a multiple case study requires more resources for data collection and handling than a single case study, which may not be always possible for practical reasons. A multiple case study design is not supposed to produce quantitative data based on the number of cases. Instead, the logic in multiple case studies is replication. A case in the study can be either literal or theoretical replication of the proposition of the study. A literal replication is a case which is expected to produce the same results as the proposition describes. A theoretical replication is a case, which is expected to produce contrasting results to the proposition, but for a previously anticipated reason. [50]

Even though a multiple case study design is recommended when it is a possibility, there are situations when a single case study design is more suitable and a multiple case study may not be even possible. Yin presents five reasons for a single case study. First reason is a critical case, which can be used when there is a well-formulated theory and a single case meets all the requirements of the theory. [50] Second reason is an extreme or unique case. This refers to a case which is so rare to come by that it is difficult or even impossible to find another case, so it is always worthwhile to investigate the case when one is found. [50, 53] Third reason is a typical case, which is considered to represent the typical company, project or whatever is the subject of investigation. Fourth reason for a single case study design is a revelatory case. This is a possibility when the investigator gets an opportunity to study a phenomenon which has not been accessible to researchers previously. Fifth reason Yin suggests is a longitudinal case, where the same case is studied at multiple different time points. [50]

A holistic case study design is a natural choice when the case does not have any recognizable subunits. However, a holistic case study design has two potential flaws. First, the case study might be on a too abstract level, if the chosen case is a large one. A holistic case study should not mean avoiding all details. The second possible pitfall of a holistic case study is that the case study may drift away from the original research questions and focus on something else that turned out to be interesting during the research. An embedded case study design, on the other hand, can provide plenty of data for extensive research. However, there is a risk that the study focuses too much on the subunits and the whole case level perspective is lost. This would effectively make the subunits the cases and the original case is just context for them. [50]

7.1.2 Case Study Protocol

The case study protocol is formed before data collection for a case study begins. It is an important factor for improving the reliability of a case study. The case study protocol is the plan for conducting the data collection for the case study. If the case study is going to have multiple investigators collecting data, they can create the case study protocol together. At least all investigators need to be familiar with the protocol. The case study protocol will help the investigator to focus on the subject of the study. In addition it requires anticipation of possible problems during the data collection beforehand.

A case study protocol should give the investigator the information they need on their site visits. There should be an overview of the case study project, e.g. the research questions, propositions and additional information about the research topic. The case study protocol should describe the field procedures, like data collection procedures and information about the site(s) to be visited. Important information can be things like accessing the site to be visited, the schedule for data collection and instructions for handling problems if they arise.

Case study questions also need to be included in the protocol. These are the questions the investigator is trying to find the answers to during the data collection, not the ones being asked from interviewees. The case study protocol should also include some information about the case study report. This can give the investigators information about the style and detail level of the report, which will help them collect data at an appropriate level and keep in mind what the data will be used for. [50]

7.1.3 Research Quality

Because case study research does not have an established framework for conducting case studies it is especially important to pay attention to the quality and credibility of the research. Yin [50] recommends using the four common tests for the quality of empirical social research as the quality criteria for a case study: construct validity, internal validity, external validity and reliability. These should be considered during the design of the case study research.

Construct Validity

Construct validity deals with the concern that the case study findings are biased or otherwise false, because subjective bias in data collection and analysis may affect the findings. The tools for improving the construct validity of a case study are using multiple sources of data, establishing a chain of evidence (see chapter 7.2.2) and having key informants of the case review the case study report.

Internal Validity

Internal validity refers to establishing causal relationships during research and is therefore only applicable to explanatory or causal types of studies. If the case study finding claims that condition A leads to condition B, i.e. condition A causes condition B, there should be strong evidence that there actually is a causal relationship and not something else. Internal validity can be improved by using pattern-matching, explanation-building and logic models in data analysis phase. Addressing rival theories is also an important tool for better internal validity.

External Validity

External validity deals with the question of whether the case study findings can be generalized to other situations except the case or cases that were investigated. In multiple case studies the replication with different cases strengthens the external validity of the research. The approach that works for both single and multiple case studies is analytical generalization. The case study research aims to generalize the results to a theory instead of another concrete case.

Reliability

Reliability is, in a way, related to the repeatability of the research. A research is considered to be reliable, if another researcher performing the same operations, like data collection and analysis, would arrive at the same findings and conclusions. Methods for improving the research reliability are the use of a case study protocol (chapter 7.1.2) and forming a case study database (see chapter 7.2.2). [50]

7.2 Data Collection

7.2.1 Sources of Data

Case study data can be collected from many different kinds of sources. Both qualitative and quantitative data can be collected. [50] Qualitative data can reveal cause or explanation for the findings from quantitative data. Quantitative data in turn can be used to support a theory suggested by qualitative data. [52] Yin presents six commonly used sources for evidence. They are documentation, archival records, interviews, direct observations, participant-observation, and physical artifacts. [50]

Documentation can range from newspaper clippings to letters and meeting minutes. The strengths of documentation are its exactness and stability. Documents can provide information from a long time frame. The problem with documentation as a data source is that the data can be intentionally or unintentionally biased by the authors. In addition the availability of documents can vary which can prevent the use of documentation completely or cause bias based on the documents which are attainable. Archival records share many of the strengths and weaknesses of documentation. The usefulness of archival records for a case study depends a lot on the case. [50]

Interviews are a popular source of data in case studies. Interviews can give a lot of useful information. An interview can focus right on the topic at hand and depending on the interviewee can provide both detailed data or focus on the bigger picture. The interviewer's ability to ask good questions is crucial. Leading questions or other personal influence can cause biased answers. [50] It is also important to choose the informants well and try to find highly knowledgeable people who view the case from different perspectives. [53]

Direct observation of events can be a great way to collect data about a phenomenon as it is happening. This produces real time data in the actual context. On the downside observation is time consuming, which makes it hard to cover a long period of time. This in turn can skew the data because only a small part of the phenomenon was observed. Observation also carries the risk that a phenomenon will proceed in a different way than it normally would because it is being observed. [50]

When the investigator no longer just passively observes but instead plays an active role in the events being studied, we are talking about participant-observation. For example a staff member in an organizational setting would be a participant-observer. Participant-observation shares the qualities of observation and adds some which are specific to this situation. The advantage of participant-observation is that it can provide insights that would otherwise be inaccessible. The disadvantage of the method is that the investigator can intentionally or unintentionally manipulate the data, which causes bias. [50]

In some case studies it is possible to use physical artifacts as data sources. When relevant for a case study, physical artifacts can provide insightful information, e.g. about cultural features. Sometimes physical artifacts, like technological devices, can store information that would not surface during a limited time site visit. Not all case studies can use physical artifacts as sources because there are no relevant artifacts to be studied. Availability of physical artifacts, even if they do exist, can also vary. [50]

7.2.2 Principles of Data Collection

Yin presents three principles of data collection, which can help improve the construct validity and reliability of the case study. These three principles should be used with all sources of data. The three principles are use of multiple sources of evidence, creating a case study database and maintaining a chain of evidence. [50]

Use of Multiple Sources of Evidence
Using multiple sources of evidence, i.e. triangulation, to support a finding makes the finding more convincing. Usually this principle refers to data triangulation, which means that multiple data sources support a finding, as opposed to making deductions based only on one data source, e.g. interview. Sometimes other triangulation types can be worth consideration as well. A good example is investigator triangulation which means that more than one investigator's evidence supports a result. [50, 52]

Creating a Case Study Database
A case study should result in two different collections of data: case study database and case study report. The case study database should include all the raw data used during the case study, e.g. documents, interview transcripts and investigator's notes. This way a critical reader does not need to blindly trust the conclusions in the case study report. They can examine the data behind the conclusions. [50]

Maintaining a Chain of Evidence
This principle also deals with the case of a critical reader. An external reader should be able to trace how the conclusions were derived from the raw data based on case study documentation. This traceability should be possible to track both ways, from data to conclusions and from conclusions to data. [50]

7.3 Data Analysis

Data analysis is one of the least established parts of case study research. There are no general purpose methods which are always useful to apply to data. This makes data analysis also one of the most difficult stages of the research. Analysis should attend to all of the evidence gathered from a case, but still maintain focus on the topic of the study. Planning data analysis early on, e.g. as a part of the case study protocol, is recommended. If analysis methods have been chosen they can be taken into account during data collection. The goal is to collect data which is analyzable. [50]

The goal of the data analysis is to build sufficient evidence that the evidence from case study research supports, or does no support, a certain theory better than any other theory. [50] If the case study is a theory building case study, the goal of data analysis is to build a theory based on the evidence. A good theory takes into account all of the available evidence. A good theory is robust and generalizable instead of being very narrow and complex. [52, 53] One strength of building theories based on case study research is that the theory is likely to be empirically valid testable. [52]

7.3.1 General Analysis Strategies

Yin [50] recommends three general analysis strategies for case studies. The first and most preferred strategy is relying on theoretical propositions. The case study usually has some preliminary propositions or theories which guide the data collection. These propositions presumably already focus on the research topic and research questions, so using them will keep the analysis also focused on the research topic.

The second general analysis strategy is thinking about rival explanations. This strategy is related to the previous one since propositions often lead to rival explanations. Rivals can still be considered even if there are no clear propositions to start with. The findings of the study become more convincing by every rival that has been ruled out. Rival explanations can be real world rivals or craft rivals. A real world rival e.g. some other intervention than the one presumed by the proposition causing an effect. Craft rivals, like investigator bias, challenge the validity of the research.

The third general strategy can be used if neither of the previous ones seems appropriate. It is developing a case description, i.e. organizing the case study using a descriptive framework. This is mainly useful for descriptive case studies.

7.3.2 Analytic Techniques

Yin [50] presents five specific analytic techniques which can be used in the data analysis of case study research. These techniques help with improving the internal and external validity of the research.

Pattern Matching

Pattern matching is the technique of comparing a pattern based on empirical information with one or several theoretical patterns. Patterns have variables which can depend on each other or not depending on a pattern. Comparing the variables of the theoretical pattern and perceived pattern determines if the patterns match or not.

Explanation Building

This technique is a special case of pattern matching and mostly only applicable to explanatory case studies. In explanation building the data analysis happens by building an explanation about the case being analyzed.

Time-series Analysis

Time-series analysis focuses on studying series of events over time. This is especially suitable approach if the case study deals with consecutive events over time; and their relationships, e.g. whether events always happen in the same order and which events lead to other events.

Logic Models

Logic models are, in a way, yet another special case of pattern matching. Logic models deal with variables which can be organized into cause-effect patterns. Cause-effect variable series can be long. A cause can also have more than one effect, which may happen at different points in time.

Cross-case Synthesis

Cross-case synthesis technique is only relevant for multiple case studies. It uses different kinds of methods for handling cross-case data. If the number of cases is large, quantitative methods may be used.

8 Research Implementation

8.1 The Case Study Project

The project which was investigated was a manufacturing execution system (MES) project in an international medical device company. The MES system is meant for both storing the records, i.e. traceability data, from manufacturing and for guiding the production worker in their work. Before the software developed during the project the same tasks have been accomplished with a combination of two or more software applications accompanied with customized MS Excel spreadsheets. The goal of the project was to improve the productivity by making the manufacturing work easier and to improve quality by adding checkups to ensure everything is going right. In addition the software needed to keep accurate and complete records of manufacturing and comply with electronic records and electronic signatures regulations [6].

The software was developed by an external software supplier. The supplier's software product was heavily modified to fit the requirements of the client. The client, not the supplier, had the responsibility of conducting formal validation to fulfill quality system requirements, as well as requirements regarding compliance with electronic records and electronic signatures [6].

The MES system was built for three production lines inside the company, each of them producing different types of products. The three production lines were each treated as separate subprojects with their own schedules and validation processes. In addition to the subprojects there were a number of common features which were also treated as their own project, to which the production line subprojects were dependent upon. At the time of writing this thesis, two of the subprojects have reached deployment stage while the third one is still being developed by the software supplier.

8.2 Data Collection Methods

Data for this thesis was collected with various different methods to get as comprehensive picture of the project and its outcome as possible. Five different types of data collection methods were used: documentation, interviews, questionnaire, observation and participant-observation.

During the research the whole project documentation created by that point in time was available. This consisted of requirements documents, validation and test plans, validation and test reports, risk analysis documents, meeting minutes and gathered user feedback. To help navigate this massive amount of documentation and to provide more insight in to the project there was the information and experience gained working as the project assistant for this particular project in the customer company since the early days of the project.

To find out more about the project and its outcome, interviews were conducted with six people who had been heavily involved in the project. They had been involved in creating and reviewing the requirements, plans and reports. They had also participated in the actual testing and deployment of the software. All of them use the software at least as a small part of their job. Four of the interviewees work as process engineers or team leaders with their work focusing on one of the production lines using the software. Two of the interviewees are from quality, regulatory and compliance department, one of whom is also the project manager of the whole project. The interviews were semi structured and had the same baseline for every interview, but the selection of exact questions depended on the interviewee. The questions handled the interviewees' opinions about the outcome, the project and the usability aspects in both of the previous ones. The interview questions (in Finnish) can be found from appendix B.

Observation and questionnaire were conducted to find out how the daily users at the production line used the software and what they thought about it. Both the observation and questionnaire were done with only one of the production lines, because that was the only production line where the software was used regularly at that point.

Most of the observation was done in one day, following what went on at the production line. During the observation many of the users were eager to explain what they were doing with the software and offered their opinions and ideas about it. All these were written down along with the observations. Together they form the observation data.

The questionnaire consisted of four different parts. The first part inquired about the user's participation in the project. The second part asked the user to rate usability related propositions on a scale of 1-7 depending on how much they agreed with the proposition. The propositions were inspired by Nielsen's usability heuristics [26], ISO 9241-11 usability definition [25], the questionnaire in the appendix of [46], ERP heuristics in chapter 6.3, AMUSE quality aspects in chapter 5.2.2 and medical device heuristics in 5.2.1. The purpose of these propositions was to find out about the usability aspects which are expected to be important in this kind of system. The third part asked the users to evaluate whether certain

tasks are easier to do with the new software or with the previous system. The last part asked the users the three best and three worst things about the software.

8.3 Data Analysis Methods

The observation data, the best and worst things from the questionnaire as well as all the remarks regarding the project outcome from the interviews were gathered to form an affinity diagram [54]. The expectation was that the affinity diagram would demonstrate the important aspects of the software and work, which had drawn attention and remarks, either good or bad. The entries were first grouped into 27 groups with a common topic. The groups were then united to form 8 larger categories.

The results from the rating usability propositions were combined. Mean, median and standard deviation values were calculated for each proposition. The comments from the interviews regarding the project were combined to find common themes.

A number of lists which contained change requests, improvement ideas and issues regarding the software were looked over. The items in the lists were categorized to see if they related to some usability aspect or regulative requirements. It was also considered how the issue might have been prevented and whether or not the issue was known before the deployment of the software.

8.4 Feasibility of Methods for Case Study Project

Many usability, and medical device usability, methods and process models are aimed at companies which create the products completely themselves, including requirements, design, implementation and testing. Since the case study project has design, implementation and some of the testing outsourced to the supplier, using these models as is does not work. Instead the general ideas of tasks, methods and principles, like the principles of human-centered design, can be applied.

During the project the customer company did not have much control over the methods used by the supplier, nor were the methods always communicated to the customer. This is why only the actions of the customer company are evaluated in this thesis. The researcher did not have much control over choosing the process model used in the project or whether or not to use usability engineering methods during the project. The time and resource constraints for the project did not leave room for anything that could be considered additional tasks.

The ERP usability methods presented in chapter 6 is better suited to the case study project, because ERP systems, just like the MES system, are often implemented by or purchased from

suppliers. This makes for example the process model represented in chapter 6.1 a good fit for the case study project, at least in theory. In addition, the ERP usability research gives suggestions which project phases may cause issues or are otherwise important from the software usability viewpoint. Both ERP and medical device usability research present suggestions about which attributes of the software system make it usable. These can be useful attributes to pay attention to also in the case study project.

The software which is the result of the case study project will be used in the manufacturing of medical devices. It will also be used as a part of the manufacturer's quality system. The software will implement electronic records and electronic signatures, as described in part 11 of code of federal regulations title 21 [6]. All of these require software validation according to quality system regulation design controls. Also the principles and tasks of software validation need to be applied to the project. Risk management is a necessary part of the case study project, as required by the quality system regulations. Managing the use-related risks is therefore a part of the project and the procedures for managing use-related risks presented in chapters 5.1.2 and 5.1.3 could be applied to the project.

8.5 The Original Hypothesis

It was expected that the tools and principles of usability engineering, especially medical device usability engineering and ERP usability engineering could, to some extent, be applied to the case study project.

No previous data directly about the effect of validation process on system usability was found. Neither was there one clear process model to be followed. Instead, there were guidelines like the principles of human-centered design. It was expected that adherence to those principles would increase the usability of the system. End user participation in the project was expected to be a very important factor for increasing the usability and user acceptance of the system. In addition to the human-centered design principles the process needed to comply with the principles of software validation presented in chapter 3.3.1 and implement the recommended software validation tasks, one way or another.

The project phases which would have major impact on how users see the system were, based on research in chapters 5 and 6, the requirements gathering and deployment. Many sources both in medical device usability and ERP usability point out the importance of requirements and design which match the user needs. The ERP usability sources bring up the importance of deployment and user training as something that will affect the users' opinions about the system a lot.

There was no previous research data to suggest that regulatory requirements would negatively affect the usability of the system. Instead it was expected that since the validation required be regulations would require a lot of effort, it might negatively affect the usability since there might not be resources available for planning and performing purely usability engineering related tasks.

9 Results

9.1 The Project

The project used process validation model as a framework for the validation activities throughout the project. In addition to the traditional validation phases a specification qualification (SQ) phase was added before design qualification. Verification and Validation (V&V) model was included in the process validation model so that specification and design qualification were treated as verification phase and operational qualification was treated as the validation phase. The validation activities were performed at system level and at subproject level. The system level activities covered the common features of the software, such as interfaces with other systems and management features. Each production line was represented by a corresponding subproject, which covered the production line specific features.

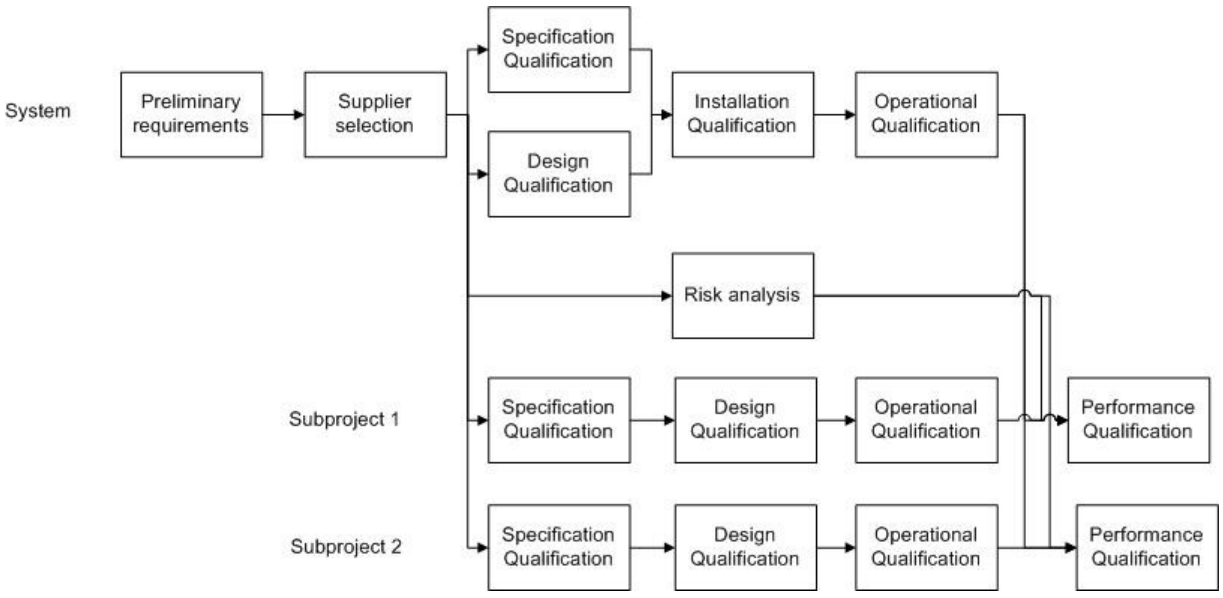


Figure 7: Project process model

Specification qualification phase for both system and subprojects was for creating more detailed requirement documents. The format used for most of the requirements was a use case. One use case represented an operation at the production, such as a test run for the manufactured product. Risk analysis for both the system in general and for the subprojects started during SQ phase.

Design qualification (DQ) for the system included technical descriptions for e.g. interfaces and hardware requirements. DQ for subprojects consisted of creating test plans and performing the first round of testing. A test plan was created for every use case in the subproject. The test plans included testing of error situations and different variations. The first round of testing was performed using the test plans. In addition, free form explorative testing was done. During subproject DQ, the software was updated regularly, first to add new features little by little and later to fix problems which were found during testing.

Installation qualification (IQ) for the system consisted of checking that the hardware and software installations meet their technical requirements. Operational qualification (OQ) for system consisted of creating and executing test plans for common system features. Procedures for technical system administration were also created.

OQ for subprojects included re-executing the testing performed in DQ, although in some cases the test plans had been updated to make testing easier. This validation testing was originally planned to be executed using the same stable software version which would then be deployed after thorough testing. In reality, software defects were still found and software updates were necessary. However, change management was much more rigorous than during DQ phase. Only small changes were made and their effect on the software quality was carefully evaluated. Additional testing was performed when it was considered necessary. After the validation testing, acceptance testing was performed. Acceptance testing consisted of virtually manufacturing a small number of products completely using the software, and replicating the manufacturing of one or more already manufactured actual products.

The subproject OQ included also activities preparing for deployment. The master data in MES system used for manufacturing was inspected. A regression testing plan for the subproject was created. User trainings were organized and new working instructions created. Procedures for system administration and manufacturing process management were created or updated.

Performance qualification (PQ) for subproject started from its deployment. Deployment started with using both the new and the old documentation side by side to make sure that MES system works, before the old system was retired and manufacturing was performed solely using the new system. PQ continued with monitoring period, which will end when the project ends, after all three subprojects have been in PQ phase long enough to build sufficient confidence that everything works.

9.1.1 Role of Usability in the Project

The project did not have formal usability activities like measurable usability requirements, usability evaluations or testing, nor did the project follow either the usability engineering process (ISO 9241) or usability engineering process for medical devices (IEC 62366). Usability was still considered during the project. One of the interviewees pointed out, that one of the reasons the software supplier was selected was that their software interface was considered likely to be simple and easy to use.

Usability issues were often written down during testing sessions. These issues were handled usually from the risk analysis point of view. Two of the most important aspects in the software were the ability to produce exact and complete traceability data from production activities and to always manufacture a product which meets its requirements. The third important aspect was efficiency, which was one of the reasons the project was originally started. Usability issues which would negatively affect any of these three aspects were considered in risk analysis sessions. Corrective actions included software modifications, configuration modifications to add more checkups, instructions and training.

Based on the interviews, usability was more prominent in this project than in most of the other software projects the interviewees had been part of. Three out of six interviewees emphasized that in this project usability improvements were made intentionally, compared to just verifying that the software works as specified, which seems to have been the case in the previous projects they have been involved in. One interviewee commented that if the effort needed to make the software work correctly and reliably had not been so big, there might have been more usability work.

9.1.2 Management of Use-related Risks

The project did not use any use-related risk specific activities but use-related risks were often handled in risk analysis meetings along with other risks. Especially once the software testing started, the testers brought up use-related hazards they had encountered while testing the system. Modifications to the system were made a number of times because of use-related risks. Still, some use-related risks surfaced only after the deployment. Since there were no pre-determined activities to specifically identify use-related hazards, finding them depended on the testers' perceptiveness and willingness to go through the trouble of reporting the potential hazards.

9.1.3 Project Feedback from the Interviews

The main topics arising from the interviews were requirements, testing, supplier quality, resources, project management and software quality at deployment. All of these topics were mentioned by more than one interviewee.

Four of the interviewees commented on requirements issues with the project. Requirements were created mostly in-house without interaction with the supplier, due to financial reasons. In some cases requirements workshops were held together with the supplier. One interviewee commented that sometimes there was ambiguity and misunderstandings with the supplier regarding the requirements. Two interviewees assumed that the complexity of the manufacturing processes and the requirements regarding the processes, traceability as well as electronic records and signatures was most likely originally not evident to the supplier. It has been also a surprise to some of the project stake holders in-house. Some requirements for manufacturing processes have surfaced after the requirements for the subproject have been written and accepted.

Test planning and testing conducted in-house received positive feedback from three persons in the interviews. The amount of testing performed during the project was huge and was naturally considered as a bit of a burden during the testing. During the interviews, all who commented testing placed it among the successful aspects of the project. So far, no critical defects have been found after the software has been deployed.

Two interviewees commented on the quality of the software released for customer testing by the supplier. There have been quite a few issues with the software releases, some of which could have been detected with more thorough testing at the supplier. There was also the issue of introducing new defects in to the software when fixing something else in a release.

Resources and scheduling received comments from two persons during the interviews. It was commented that the expectations from the upper management were not always realistic in this area. The workload was estimated to be much smaller than it turned out to be, which affected the schedule. The deployment for the first subproject was over a year late. Most people were doing the project alongside with their normal work, which caused stress and conflicting requests for their time.

Project management received remarks from two persons in the interviews and it was considered to be one of the successes of the project. It was commented that a project needs a clear leadership and a stable group of people who are committed to the project. A clear vision of what is to be created is necessary and the vision needs to be communicated clearly.

Three interviewees considered it a successful choice, that the software was not deployed before it was ready for it, despite the noticeable delay from the planned schedule. These interviewees felt that the reception at the manufacturing lines would have been far more negative, if the software had been deployed in a less mature state. Negative first impression might have permanently affected the attitudes towards the software.

9.1.4 Adherence to the Principles of Human-centered Design

9.1.4.1 The Design is Based upon an Explicit Understanding of Users, Tasks and Environments

Although the end users did not participate in the project in a big way, they were considered pretty well during the project. No major deficiencies were uncovered during the research. The stakeholders who are not main users were considered less, but not completely forgotten. For example the people who insert information to the ERP system, which MES reads, were considered and involved, because the importance of correctly formulated ERP system data surfaced during testing. On the other hand people who might just need to access some small part of data on MES were considered less.

The only issue uncovered during research regarding environment or context of use, was not an issue actually depending on the MES software and more related to the question of where and with what kind of device the system should be used. This issue had been considered during the project, but resource limitations affected the possible solutions.

The understanding of users' tasks was achieved in the end and the deployed software supports users in their tasks. However, the road there might have been a bit shorter. During the project there were quite a few moments when a completely new requirement appeared during testing or test planning, or it was discovered that a practice which affected the requirements had changed some time ago and the information had not reached the project in time.

9.1.4.2 Users are Involved throughout Design and Development

The super users, who have become the administrators of the production related data, have been involved in the project activities after the supplier selection. The production workers, who are the daily end users, have participated less. Especially the requirements gathering phase and early testing phases could have benefited from more frequent end user feedback. As has been mentioned in the previous section, the understanding of users' tasks and processes could have been better. Having end users participate in the requirements gathering

phase would likely have lessened deficiencies in requirements. The next section describes the participation of users in evaluation activities.

When asked in the questionnaire, 6 of 15 production workers replied that they would have wanted to participate more in the MES system project. Eight respondents would not have wanted to participate more and one questionnaire had no answer to the question.

When asked to clarify how they would have wanted to participate more, the following answers were given (some respondents had more than one comment to this): Two respondents would have wanted to be more involved in fitting the MES system to the work, so that they would match better together. One person would have liked to participate in testing the system. One would have wanted to be involved in the details of the user interface that they have to use most. One respondent would have wanted to see the system before the final version, and presumably be able to comment on it. One person would have liked to be able to go through different kinds of cases using the system; another would generally have wanted to be more involved with the project. In general, the people who would have wanted to participate more seem to wish they could have affected what the system became like instead of just be given a finished system to use.

9.1.4.3 The Design is Driven and Refined by User-centered Evaluation

The super users of the system were heavily involved in the testing phases giving them plenty of opportunities to give feedback about the system design. No formal user-centered evaluation methods, like usability testing were used due to limited resources. The following paragraphs describe how the production workers, who are the main daily end users of the system, participated in the evaluation of the system.

The first couple of releases from the supplier were accompanied with a presentation session where the supplier would present the new features to project team members and end user representatives. The presentations were also a venue for giving feedback and asking questions about the design.

There were also some end users involved in the testing activities, during which their comments were collected and processed afterwards. The end user participation in testing was divided unevenly among the subprojects. Subproject 1 had all except one of their production workers participate in testing. Subproject 2 had just a production team leader representing the production workers in testing sessions. The reason for differing participation levels was different needs for workers in production during testing. The only exception to production worker participation was performance testing session, which had production workers from both subprojects participating to create enough load to the system. The production workers

had a chance to comment on the system during and after the performance testing session, but they were not given a full tour of the system or given tasks that simulate the actual use situation. There were not many comments after the testing session, except for unexpected error messages given by the system, which were mostly caused by still existing software defects.

During and after deployment user comments and feedback was collected to a list. This list was processed by the project team to decide on measures for each entry. Some of the comments are on the feature lists for future software updates; some will be addressed in-house by altering the system configuration. Some comments will lead to no further actions, because the issue is considered to be minor or because fixing the reported issue would violate regulatory requirements.

9.1.4.4 The Process is Iterative

Since the software was designed and developed by the supplier, it was not possible to influence the design process. Design and implementation for new features was often completed by the supplier as a whole. Sometimes there was a design document that was approved by the project team before implementation, sometimes there was not. After implementation improvements were made mostly to fix defects.

On the other hand the project as a whole was iterative. The software was developed in sprints, releasing features for testing little by little. There were two full rounds of formal testing for each subproject, during which the software was improved.

9.1.4.5 The Design Addresses the Whole User Experience

The project approached usability and user experience from the practical viewpoint. Making the use of the system exciting or fulfilling was not important. MES is a mission-critical system, for which effectiveness and efficiency outweigh user satisfaction. Both efficiency and user satisfaction might have improved if MES supported every single task for production workers, but resource limitations forced to make some compromises with this.

9.1.4.6 The Design Team Includes Multidisciplinary Skills and Perspectives

The project had a small team of people who were heavily involved in the project. This team included a couple of representatives for each production line, a couple of people from the quality, regulatory and compliance (QRC) department, and the project assistant, who had background in software engineering and human-centered design. Depending on a matter at

hand, relevant team members were joined by supplier's software developers or in-house experts of e.g. system administration, sales or purchasing.

The testing performed by the customer was always done with at least two people participating in the testing session. Most of the time one of these people would have their background in quality and regulatory affairs, and the other in manufacturing. Although this took a lot of human resources, it was usually a good solution. People with different backgrounds would look at the system from a different point of view and two people would spot more noteworthy issues than one person.

Since MES directly affects the quality system compliance of the company, it was both useful and necessary to have a strong QRC representation in the project team. Having production worker end users involved in the project early on and working more closely with the supplier during requirements gathering and design might have made the project easier and improved the result.

9.2 Usability of the System

9.2.1 Usability Attributes

From observation data, comments from production workers and interview comments of project team members 8 categories were formed, which all include more than one smaller topic. These 8 categories are presented below. Some of the topics as well as single data entries could belong to more than one of the larger categories, which is why no deductions have been made on the importance of each of the categories based on the number of topics or data entries.

Correctness of User's Actions
Correctness of user's actions means the absence of errors in the work. The topics that contribute to correctness of actions include e.g. error prevention, sufficient guidance, standardized work practices and correctness of instructions.

Understandability
Understandability refers to the user's ability to understand how the system works. The topics contributing to understandability are complexity of the system, consistency of the system operating logic, learnability and how confident the user feels about using the system.

Accessibility of Information

Accessibility of information describes how easy it is for a user to find or notice the information they need. The topics contributing to accessibility of information are e.g. searching for information, visibility of status and flow of information.

Unnecessary Work

Unnecessary work refers to work that the user feels is pointless and does not contribute to their actual work goals. The topics contributing to this category are mostly variations of re-inserting data in to the system, which from the users' point of view the system should already remember. Having to repeat a set of actions because the system tools do not support handling larger bits of data was also a minor theme.

Suitability for the Work

Suitability for the work describes the system's ability to support the concrete work. The topics contributing to suitability are match between the system and the work and the completeness of the system in covering the work process, i.e. are there actions that need to be taken outside the system.

Correctness of Records

Correctness of records refers to the exactness, correctness and completeness of the records, which store the manufacturing history of products. The topics which contribute to the correctness of records are traceability, real-time recording of events in to system and fixing errors.

Efficiency

Efficiency refers to the system's ability to support the user in getting more work done in smaller amount of time. The topics which contribute to efficiency are the amount of paperwork, having one system instead of many; and speed and fluency of work.

Comfort

Comfort describes how comfortable it is for a user to use the system. The topics which contribute to comfort of the system are the visual amenity of the interface, ergonomics of the work with the system and the flexibility of the system.

Most of these usability attributes or similar quality descriptors can be found from the usability attributes and heuristics in the medical device usability and ERP usability research presented in chapters 5 and 6.

9.2.2 Users' Ratings on System Usability

Table 1 presents the results of end users rating statements about the MES system based on how much they agree with the statement. The possible range for replies was from 1 to 7, number 7 being the highest rating to the MES system and number 1 the lowest. The overall mean value from all ratings is 4,59 which is a little above the average of 4. The most favorable ratings suggest that the production workers find MES system fairly easy to learn and use. The MES system is also regarded as quite trustworthy and pleasant to use. On the other end of the scale, it is easy to see, that according to the production workers, once a mistake has happened, it is not easy to recover from it using MES.

Table 1: System statement questionnaire results

Statement	Mean*	Standard variation	Median*
It is difficult to use MES.	5,27	1,71	6
I learned easily to use MES.	5,73	1,03	6
I usually fail if I try to use a new or rarely used MES functionality.	5,33	1,45	6
Using MES has slowed down my working pace.	4,20	2,27	5
Working using MES is efficient.	4,67	1,45	5
MES guides me enough in my work.	4,87	1,41	5
I need to memorize things to be able to use MES.	3,93	1,58	4
It is easy to understand MES error messages.	4,33	1,84	5
If I make a mistake, fixing it using MES is difficult.	2,00	1,51	1
It is unpleasant to use MES.	5,33	1,54	6
Using MES is a natural part of my work.	4,93	1,49	5
I trust that MES works as it should.	5,20	1,70	5
It is easy to make mistakes while using MES.	4,33	1,50	4
I find the information I need easily from MES.	4,47	1,68	5
MES uses misleading words or terminology.	4,60	1,59	4
Using MES has increased my productivity.	4,00	1,77	4
MES is useful to me.	4,87	1,46	5

* Numbers adjusted so that number 7 is always the most favorable rating to the MES system, number 1 is the worst.

Another part of the questionnaire asked the production workers to evaluate whether certain tasks are easier to perform using the new MES system or using the old combination of systems and Excel spreadsheets. The possible range for replies was from 1 to 7, number 1 meaning easier with the new system and number 7 easier with the old system. Table 2 presents the mean, standard deviation and median values for those tasks that gathered more than three replies. Users were instructed to only evaluate the tasks they have performed themselves using both systems.

The overall mean value for the replies, including the tasks omitted from the table, was 4,64 slightly in favor of the new MES system. Starting a work order and running tests during manufacturing were the tasks where the new system was most preferred over the old one. Starting a work order was known beforehand to be time consuming and unnecessarily complex before the new system, so this result is not surprising. Settings, changing a part and creating an error report were the tasks where the users slightly preferred the old system to the new one. Settings are a task which has little support in the new MES system, which probably explains the low rating. Changing a part and creating an error report are related to recovering from error situations, which already got low ratings in the MES system in table 1.

Table 2: Comparison questionnaire results

Task	Mean**	Standard deviation	Median**
Starting a work order	6,78	0,44	7
Assembly	4,33	1,32	4
Settings	3,5	1,27	4
Running tests during manufacturing	5,25	1,04	6
Changing a part	3,6	1,78	4
Creating an error report	3,91	1,87	5
Re-running a test	4,11	2,03	4
Searching for information	4,83	1,70	5

** Number 1 means the task is easier in the previous system; number 7 means the task is easier in the new MES system

When the production workers were asked to list the best and worst qualities of the MES system, the positive comments were mostly related to efficiency. The tasks that are easy and fast to perform with the new system were often mentioned. Some liked also the availability of information and a few people mentioned liking that the system guides the worker and that there are checklists which help prevent errors.

A large number of the worst qualities were also related to efficiency. Cases which require re-inserting information were disliked as well as the tasks that are considered slow with the new system. Recovery from errors was considered too difficult and time consuming. In addition to

efficiency, quite many negative comments related to the system not matching or supporting the actual work well enough. A few comments related to the inflexibility of the system in some situations. The reason for some of the negative comments about flexibility and work support may be that some of the workers needed to change their working routine. The new system forces everyone to use the same processes, which makes it easier to control quality.

9.3 Issues and Improvement ideas

9.3.1 Change Requests

Change requests are issues which require an update to the software system. The change requests are either clearly outside the scope of the original system requirements or it is unclear whether they are covered by the requirements or not.

The latest change request list has 24 change requests. 10 of those have been reported already before deployment for either subproject. Two out of 24 have already been implemented and will be deployed in the next software update. Additional 13 change requests have been listed previously, which are no longer included in the latest change request list. Three of the thirteen older change requests have been implemented; other 10 have been discarded either as outdated or not feasible in the near future.

Out of the 24 remaining change requests, all except three can be classified as usability improvements. The three other change requests are related to regulatory requirements. At least three of the usability related change requests are also related to regulatory requirements. The regulatory requirements related change requests are mostly improvements for gathering more detailed traceability data or improvements to preventing and recovering from error situations. Usability related change requests mostly belong to one or more of three categories: easy access to information, efficiency and dealing with error situations.

9.3.2 QRT Issues

Quick Response Team (QRT) is a team which handles the recorded issues during the software validation, both before and after the deployment. The QRT has members from quality and regulatory, manufacturing and MES system project.

There are 19 issues which have been handled by the QRT during the follow up period after the deployment of the second subproject. Two of these issues are not directly related to the MES software. The remaining 17 issues all are at least partially usability related. Most of the issues belong to one or more of the following categories: missing, misleading or erroneous

information, efficiency, something not working, system not being able to prevent use error or allow recovery from it and system not working as expected by the user.

Out of the 17 usability related issues only one was known before the deployment, but the consequences of the issue were not investigated thoroughly. The issue has later been mended by the implementation of one of the change requests. Four of the remaining 16 issues were rare or random so that it would have been quite difficult to reliably catch them before deployment. Most of the 12 remaining issues should have been caught during either design or testing. Many of the issues arise from the same situation: A feature has been designed by the supplier to fulfill the letter of the requirements written by the customer. However, the consequences of the design solution are causing unwanted effects. This is probably caused by the combination of the supplier not being familiar enough with the users' work to understand that the design might cause problems; and the customer not understanding the consequences of the design solution. More dialogue between the supplier and customer might have been needed during the design phase. The testing by the customer could have caught these issues, but in many cases that would have required the involvement of the actual users as well as use of more realistic data and use situations.

The QRT records include two issues in which a feature aimed at improving the quality and control of the manufactured products, i.e. improving the compliance with regulatory requirements, adversely contributed to the usability of the MES system. One of these issues caused sometimes unnecessary manual tasks to the users. The other made it impossible to recover from a use error. It was possible to prevent the latter issue appearing again by changing the configuration of the system.

9.3.3 Improvement Suggestions

Improvement suggestions were collected and recorded approximately during the first three months of the system usage. The improvement suggestions come directly from the different users of the MES system: the production workers, quality and regulatory as well as the administrators, who use the system to manage the manufacturing processes.

Total of 54 improvement suggestions were recorded during the follow-up period. These include a couple of overlapping suggestions, but not many. 22 of the 54 suggestions are not directly commenting on the features of the system, instead they can and mostly have been fixed by updating the configuration of the manufacturing processes in the system or by updating the work instructions. These improvements have improved the match between the system and the actual work.

Out of the remaining 32 suggestions, two are not usability related, leaving 30 usability related improvement suggestions to system functionalities. Over one third of these are mostly minor efficiency issues i.e. things that the users feel could be automated or made otherwise easier. Other recurring themes are the easy availability of needed information, matching the system better with the actual work, decreasing users' mental load as well as preventing and recovering from errors.

16 of the 30 usability related issues were already known before the deployment. As with QRT issues, most of the improvement suggestions that the project team was not aware of before deployment, could have been found during the design or testing phases. Having more actual end users and more realistic use scenarios could have revealed some of these issues earlier.

Seven of the usability related issues are related to regulatory requirements. A couple of these are directly related to the compliance with the electronic records and electronic signatures requirements causing additional work. Other cases have to do with ensuring the completeness and correctness of manufacturing traceability data and ensuring independence of review, i.e. ensuring no one reviews their own work.

10 Discussion and Conclusions

10.1 The Research Questions

10.1.1 How does the Validation Process for Medical Device Software Affect the Usability of the Software?

There was no clear evidence of the medical device validation process directly affecting the usability of the software. The validation process does not need to include usability specific activities, but on the other hand, they can be integrated to be part of the process if usability focus is chosen.

The validation process can be used to help improve the usability aspects of the software, as was done in the case study project. The extensive documentation required from validation can be used to improve the usability of the software. The documentation can be used to add usability concerns to requirements, from where they will be transferred to validation and test plans. If the software design and development is outsourced to a supplier, it is especially important to integrate usability goals in to the requirements. The research data from the case study suggests that efficiency, easy access to relevant information and proper handling of error situations along with sufficient support to the end users' tasks are important usability factors.

Risk management needs to be part of the validation process and use-related risks need to be managed as well as the other risks. A sufficient focus on use-related risks will help with diminishing the likelihood of having unexpected hazardous features in the final software.

10.1.1.1 Is Usability in Some Situations Over Ruled by Regulatory Requirements?

The expectation was that regulatory requirements would only affect usability indirectly, i.e. the resources needed to comply with regulatory requirements might be away from the resources which could be used to improve usability. This turned out to be true in the case study project: there were not enough resources to perform tasks solely aimed at improving usability.

In addition it was discovered that in some cases regulatory requirements will over rule usability of the software. In the case study project there were some system features that the

users would have liked to change to make things faster or more flexible. In some cases the users might not know why the improvement they suggest cannot be implemented, because they are not intimately familiar with the details of the requirements.

10.1.2 How can we Produce Usable, Medical Device Regulated Software?

10.1.2.1 How Should the End Users be Involved in the Validation Process?

The experiences from the case study project support the claim that end users should be involved in the project from start to finish. It would be especially beneficial to involve the end users in the early phases of the project to make sure that the requirements and design decisions will meet the end user needs. End user input is needed for requirements, design and evaluation phases of the software project. The earlier the input can be obtained, the easier it will be to use the input to influence the project outcome.

The methods for describing how the end users should be involved in a project are also important. The case study did not use any specific methods; instead the users were participants in the testing sessions or meetings like everyone else. However, based on both the experiences from testing sessions, and the user feedback in questionnaires and improvement suggestions, it is clear that usage of some other methods might be beneficiary. Especially if the end users are not constantly part of the project they may not have the whole picture, nor are they always aware of regulatory requirements. This combined with direct end user input may lead to inefficient design solutions, when users comment based on their current work practices instead of the new ones, or non-complying software. Using usability engineering methods, like contextual inquiry, observation and usability testing would probably be a good addition for obtaining information from end users.

10.1.2.2 What Kind of Methods should be used, in which Stage of the Process and by Whom?

The previous chapter already discusses the participation of end users and the methods that should be used to obtain end user feedback. To properly use the usability engineering methods suggested in the previous chapter, a usability engineering knowledge is needed in the project.

It has been a good solution to have a multidisciplinary project team, as is suggested by the principles of human-centered design. If the software that is developed during the project is going to be strongly linked to the quality management and compliance with regulatory requirements, as was the situation with the case study project, it is recommended to have project team members with quality and regulatory knowledge. For a software system like this one, the ideal solution takes in to account both the quality and the manufacturing needs.

A method used in the case study project that can be recommended was testing sessions with more than one person, typically two. Two people have the ability to notice more important things than one person. Also having people with different backgrounds, typically quality and manufacturing, inspect the system brought up a variety of noteworthy issues that might otherwise have been left unnoticed before deployment.

There should be a method for iteratively evaluating design decisions before they are set in stone. The case study project had some issues with design decisions being made based solely on written requirements with little or no discussion between the supplier and customer representatives. It often happened that the requirement was misunderstood or that the design decision brought up more requirements or risks that needed to be addressed. Both quality and end user representatives, as well as necessary subject matter experts, should be part of the iterative design.

10.1.2.3 What the Software Validation Process should Look like for Medical Device Regulated, Usable Software?

Based on this one case study project it is impossible to suggest one process model which will always work. Instead suggestions can be given on the attributes that the process model should have. The process should naturally adhere to the principles of software validation in chapter 3.3.1 and the quality system regulation [2]. The principles of software validation already present activities which should be included in the software validation process (see chapter 3.3.2). The rest of this chapter focuses on the other desirable aspects of the validation process.

Iterative process structure is recommended like in the principles of human-centered design. The requirements and design are bound to change a little during the project. Seeing design decisions will also inevitably bring up comments and requirements which might otherwise remain hidden. Iterative process makes it possible to react to these changes.

End user involvement has already been discussed in chapter 10.1.2.1. Risk management should be a part of the validation starting from the very beginning. Risk management activities should continue through the project, including recognizing new risks and evaluating the old ones and their control measures. It is important to pay sufficient attention specifically to use-related risks.

10.1.2.4 Which are the Critical Points in the Validation Process?

As was expected, the early phases including defining requirements and design decisions had far-reaching consequences in the project. Thorough testing was also considered a major

factor in ensuring that the developed system will be successful. It was considered important to the user acceptance of the system, that the software was not deployed until it was mature enough and the biggest rough edges had been smoothed out.

There was no indication that the actions during deployment, like user training, would have had a big effect on the acceptability of the system. However, the deployment including trainings was considered successful; the importance of the deployment might have been more evident had there been problems with it. Instead it was considered that presenting the project in positive light to the end users who were not actively involved in the project had a huge influence on their opinions about the system.

As was mentioned in a couple of previous studies on medical device usability, management support to usability seems to be important. The project team may want to focus on usability, but with tight schedules and limited resources there is not much that can be done.

10.2 Research Assessment

The choice to conduct a single case study was somewhat acceptable, since it would have been difficult to find another project with a sufficiently similar setting. With only one researcher, it would also have been quite impossible to gather the participant-observation data similarly from other projects. However, to draw more convincing conclusions, more cases would have been needed, or the selected one case could have provided more useful data, had it been a bit different. Especially the method suggestions which were not used in the case study project should be studied, possibly with another case study which experiments with one or more of them.

The case study project in this thesis had potential to be an embedded case study with its' multiple subprojects. They could have provided a variety of data and comparisons could have been made between subprojects. This was not possible in practice, since the first deployed subproject did not use the new system much during the research period and the third subproject was in a very early stage. Therefore most of the data came from only one subproject.

Construct validity and reliability of the research are quite good. Multiple sources of data have been used and original data, participant-observation information excluded, is still available. Internal and external validity of the research could be better. Since this was a theory building case study, there was not a clear theory to begin with and even less plausible rival theories to be addressed. The results do not form a concise and complete theory. On the other hand,

some of the findings can be generalized as common instructions; in fact many suggestions made based on case study findings can be found from existing usability engineering methods.

10.3 Future Work

More research is needed to strengthen the findings of this thesis. More cases would need to be studied to test the results. Especially interesting would be the possibility to study projects which use usability engineering methods and principles in the requirements gathering phase to see if it makes a difference.

References

1. CHENG, M. *Medical Device Regulations: Global Overview and Guiding Principles.* World Health Organization, 2003 ISBN 9241546182.

2. Food and Drug Administration. *Code of Federal Regulations Title 21 Part 820 Quality System Regulation.* U.S. Department of Health and Human Services. 09/01/2014, 2014 Available from:http://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?CFRPart=820.

3. Food and Drug Administration. *General Principles of Software Validation; Final Guidance for Industry and FDA Staff.* , January 11, 2002.

4. ACHARYA, C., THIMBLEBY, H. and OLADIMEJI, P. *Human Computer Interaction and Medical Devices.* Dundee, United Kingdom ed. Swinton, UK, UK: British Computer Society, 2010 ISBN 978-1-78017-130-2.

5. KAYE, R. and CROWLEY, J. *Medical Device use-Safety: Incorporating Human Factors Engineering into Risk Management.* U.S. Department of Health and Human Services, FDA, July 18, 2000.

6. Food and Drug Administration. *Code of Federal Regulations Title 21 Part 11 Electronic Records; Electronic Signatures.* , 2015.

7. *Council Directive of 20 June 1990 on the Approximation of the Laws of the Member States Relating to Active Implantable Medical Devices (90/385/EEC).* , 1990 ISBN OJ L 189, 20.7.1990, p. 17.

8. *Council Directive 93/42/EEC of 14 June 1993 Concerning Medical Devices.* , 1993 ISBN OJ L 169, 12.7.1993, p. 1.

9. *Directive 98/79/EC of the European Parliament and of the Council of 27 October 1998 on in Vitro Diagnostic Medical Devices.* , 1998 ISBN Official Journal L 331 , 07/12/1998 P. 0001 - 0037.

10. SFS-EN ISO 14971. *Medical Devices. Application of Risk Management to Medical Devices.* , 2009.

11. SFS-EN ISO 9000. *Quality Management Systems. Fundamentals and Vocabulary.* , 2005.

12. SFS-EN ISO 13485. *Medical Devices. Quality Management Systems. Requirements for Regulatory Purposes.* , 2012.

13. SUDERSHANA, S., VILLCA-ROQUE, A. and BALDANZA, J. *Successful Collaborative Software Projects for Medical Devices in an FDA Regulated Environment: Myth Or Reality?* , 2007 DOI 10.1109/ICGSE.2007.40.

14. CURTS, C. FDA-Regulated Validation in Clinical and Nonclinical Environments (Regulatory Affairs). *Engineering in Medicine and Biology Magazine, IEEE*, 2007, vol. 26, no. 1. pp. 91-97 ISSN 0739-5175.

15. ABDEEN, M.M., KAHL, W. and MAIBAUM, T. *FDA: Between Process and Product Evaluation.* , 2007 DOI 10.1109/HCMDSS-MDPnP.2007.6.

16. The Global Harmonization Task Force - SG3. *Quality Management Systems - Process Validation Guidance.* 2nd ed. , 2004.

17. Validation Online. *Design Qualification (DQ).* Available from:http://www.validation-online.net/design-qualification.html.

18. Working Party on Control of Medicines and Inspections. *Revised Draft Qualification and Validation.* European Commission, 2000.

19. PRESSMAN, R.S. *Software Engineering : A Practitioner's Approach.* 6. ed. ed. New York: McGraw-Hill, cop. 2005 ISBN 0-07-285318-2.

20. NABIL MOHAMMED, A.M. A Comparison between Five Models of Software Engineering. *International Journal of Computer Science Issues*, 2010, vol. 7, no. 5. pp. 94-101 ISSN 1694-0784.

21. ANITHA, P.C., SAVIO, D. and MANI, V.S. *Managing Requirements Volatility while "Scrumming" within the V-Model.* , 2013 DOI 10.1109/EmpiRE.2013.6615211.

22. Food and Drug Administration. Guidance for Industry, FDA Reviewers and Compliance on Off-the-Shelf Software use in Medical Devices, 1999.

23. FAULKNER, X. *Usability Engineering.* Basingstoke: Palgrave, 2000 ISBN 0-333-77321-7 (nid.).

24. ISO 9241-210. *Ergonomics of Human-System Interaction -- Part 210: Human-Centred Design for Interactive Systems.* , 2010.

25. ISO 9241-11. *Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs) -- Part 11: Guidance on Usability.* , 1998.

26. NIELSEN, J. *Usability Engineering.* San Francisco (CA): Academic Press, 1993 ISBN 0-12-518406-9.

27. LEVENTHAL, L.M. and BARNES, J.A. *Usability Engineering : Process, Products, and Examples.* Upper Saddle River, N.J: Pearson/Prentice Hall, cop. 2008 ISBN 978-0-13-157008-5.

28. BEYER, H. and HOLTZBLATT, K. Contextual Design. *Interactions*, jan, 1999, vol. 6, no. 1. pp. 32-42 ISSN 1072-5520.

29. IEC 62366. *Medical Devices -- Application of Usability Engineering to Medical Devices.* , 2007.

30. PATTERSON, P.A. and NORTH, R.A. Fitting Human Factors in the Product Development Process. *Medical Device & Diagnostic Industry*, 2006, vol. 28, no. 1.

31. BRAUN, S. *Usability for Medical Devices.* , 2005 DOI 10.1109/PSES.2005.1529517.

32. VIITANEN, J. and NIEMINEN, M. Terveydenhuollon Tietojärjestelmien Käytettävyys. *Finnish Journal of eHealth and eWelfare*, 2009, vol. 1, no. 3 ISSN 1798-0798.

33. VAN DER PEIJL, J., KLEIN, J., GRASS, C. and FREUDENTHAL, A. Design for Risk Control: The Role of Usability Engineering in the Management of use-Related Risks. *Journal of Biomedical Informatics*, 8, 2012, vol. 45, no. 4. pp. 795-812 ISSN 1532-0464.

34. ISO 60601-1-6. *Medical Electrical Equipment – General Requirements for Basic Safety and Essential Performance – Collateral Standard: Usability.* , 2006.

35. DOERR, J., et al. *Supporting Requirements Engineering for Medical Products: Early Consideration of User-Perceived Quality.* Leipzig, Germany ed. New York, NY, USA: ACM, 2008 ISBN 978-1-60558-079-1. DOI 10.1145/1368088.1368177.

36. THIMBLEBY, H. *User-Centered Methods are Insufficient for Safety Critical Systems.* , 2007 SCOPUS.

37. ZHANG, J., et al. Using Usability Heuristics to Evaluate Patient Safety of Medical Devices. *Journal of Biomedical Informatics*, 0, 2003, vol. 36, no. 1–2. pp. 23-30 ISSN 1532-0464.

38. ISO/IEC 9126-4. *Software Engineering - Product Quality - Part 4: Quality in use Metrics.* , 2004.

39. DAVIS, F.D. Perceived Usefulness, Perceived Ease of use, and User Acceptance of Information Technology. *MIS Quarterly*, Sep., 1989, vol. 13, no. 3. pp. 319-340 ISSN 02767783.

40. DOERR, J., et al. *Built-in User Satisfaction - Feature Appraisal and Prioritization with AMUSE.* , 2007 ISBN 1090-705X. DOI 10.1109/RE.2007.62.

41. SCHOLTZ, B., CILLIERS, C. and CALITZ, A. *Qualitative Techniques for Evaluating Enterprise Resource Planning (ERP) User Interfaces.* Bela Bela, South Africa ed. New York, NY, USA: ACM, 2010 ISBN 978-1-60558-950-3. DOI 10.1145/1899503.1899535.

42. SINGH, A. and WESSON, J. *Evaluation Criteria for Assessing the Usability of ERP Systems.* Vanderbijlpark, Emfuleni, South Africa ed. New York, NY, USA: ACM, 2009 ISBN 978-1-60558-643-4. DOI 10.1145/1632149.1632162.

43. PARKS, N.E. *Testing & Quantifying ERP Usability.* Calgary, Alberta, Canada ed. New York, NY, USA: ACM, 2012 ISBN 978-1-4503-1643-9. DOI 10.1145/2380790.2380799.

44. TOPI, H., LUCAS, W. and BABAIAN, T. *Identifying Usability Issues with an ERP Implementation.* , 2005 SCOPUS.

45. VILPOLA, I. and VÄÄNÄNEN-VAINIO-MATTILA, K. *Evaluating Human Factors in ERP Implementations.* , 2005.

46. CALISIR, F. and CALISIR, F. The Relation of Interface Usability Characteristics, Perceived Usefulness, and Perceived Ease of use to End-User Satisfaction with Enterprise Resource Planning (ERP) Systems. *Computers in Human Behavior*, 7, 2004, vol. 20, no. 4. pp. 505-515 ISSN 0747-5632.

47. VILPOLA, I.H. A Method for Improving ERP Implementation Success by the Principles and Process of User-Centred Design. *Enterprise Information Systems*, 02, 2008, vol. 2, no. 1. pp. 47-76 ISSN 17517575.

48. MÄKIPÄÄ, M. *Toiminnanohjausjärjestelmän Käyttöönotto – Teoreettinen Metodi Ja Koettelu Kahdessa Case-Yrityksessä.* Tampereen yliopisto, 2002.

49. VILPOLA, I., VÄÄNÄNEN-VAINIO-MATTILA, K. and SALMIMAA, T. *Applying Contextual Design to ERP System Implementation.* Montreal, Quebec, Canada ed. New York, NY, USA: ACM, 2006 ISBN 1-59593-298-4. DOI 10.1145/1125451.1125485.

50. YIN, R.K. *Case Study Research : Design and Methods.* 3rd ed. ed. Thousand Oaks (CA): Sage Publications, cop. 2003 ISBN 0-7619-2552-X.

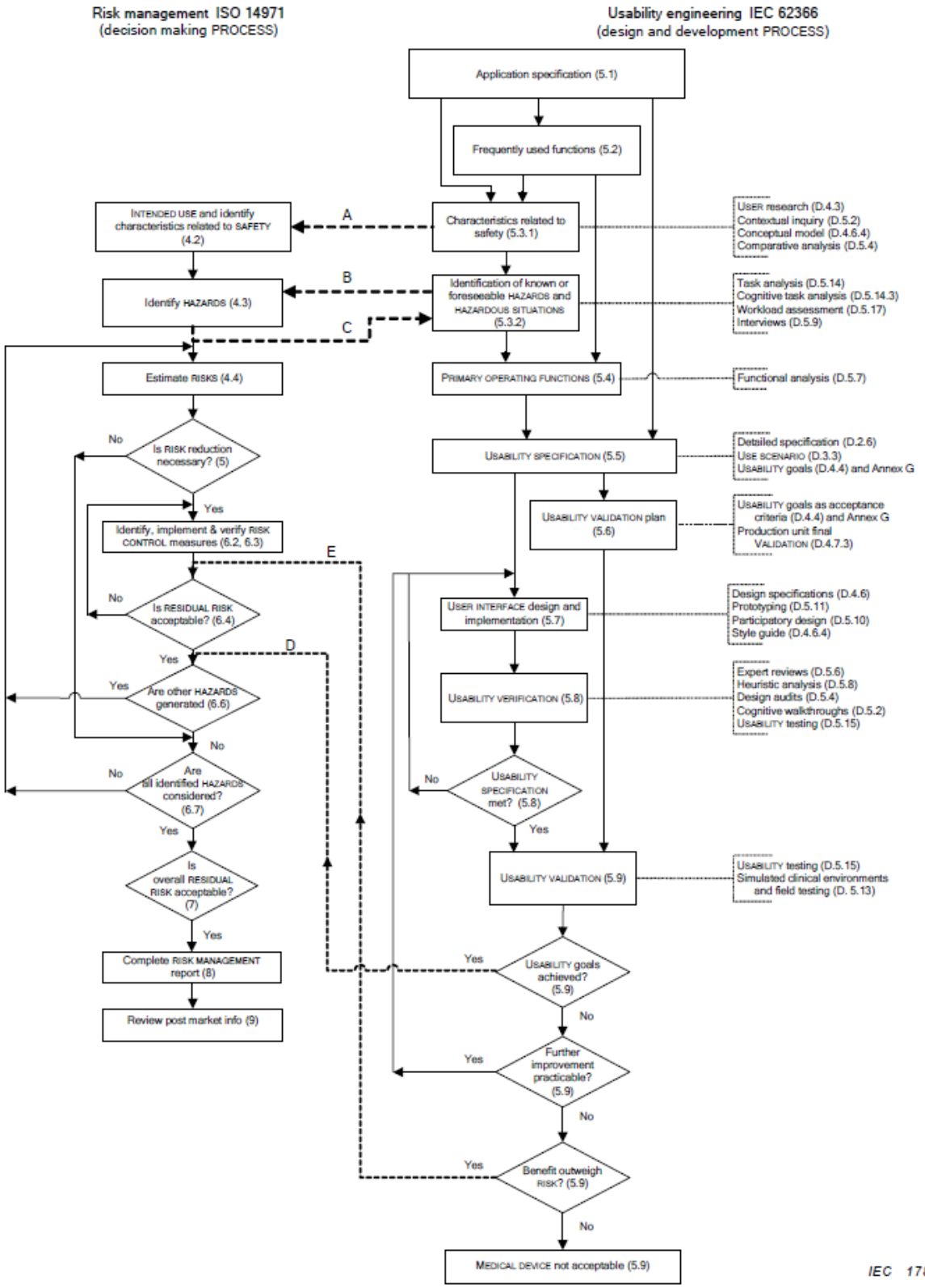
51. BENBASAT, I., GOLDSTEIN, D.K. and MEAD, M. The Case Research Strategy in Studies of Information Systems. *MIS Quarterly*, Sep., 1987, vol. 11, no. 3. pp. 369-386 ISSN 02767783.

52. EISENHARDT, K.M. Building Theories from Case Study Research. *Academy of Management Review*, 10, 1989, vol. 14, no. 4. pp. 532-550 ISSN 03637425.

53. EISENHARDT, K.M. and GRAEBNER, M.E. Theory Building from Cases: Opportunities and Challenges. *Academy of Management Journal*, 02, 2007, vol. 50, no. 1. pp. 25-32 ISSN 00014273.

54. BEYER, H. and HOLTZBLATT, K. *Contextual Design : Defining Customer-Centered Systems.* San Francisco (CA): Morgan Kaufmann, 1998 ISBN 1-55860-411-1 (nid.).

Appendix A: Information Flow between ISO 14971 and IEC 62366 Processes



Appendix B: Interview Questions in Finnish

Miten käytät järjestelmää? Minkälaisia asioita teet järjestelmällä? Kuinka usein?

Miten hyvin järjestelmä palvelee omia tarpeitasi? Entä miten hyvin se sinun näkökulmastasi palvelee tuotannon tarpeita?

Vertailu järjestelmää edeltävään aikaan (sekä oma työ että tuotannon sujuminen yleisesti): Mitkä asiat ovat vaikeutuneet, mitkä helpottuneet? Muita merkittäviä muutoksia?

Oma roolisi projektissa? Mihin osallistuit? Milloin mukaan projektiin? Vaikutusmahdollisuudet?

Kun ajatellaan koko projektia toimittajavalinnasta vaatimusten ja testausten kautta käyttöönottoon: Mitkä asiat projektissa ovat mielestäsi onnistuneet? Mitä olisi kannattanut tehdä eri tavalla ja miten?

Oletko ollut mukana muissa samantyyppisissä validointiprojekteissa yrityksessä? Mitä eroja ja yhtäläisyyksiä tähän projektiin verrattuna?

Miten projektissa omasta näkökulmastasi otettiin käytettävyysasioita huomioon? Mikä onnistui, mitä olisi pitänyt tehdä toisin?

Jos verrataan projektia muihin samantyyppisiin yrityksen projekteihin, onko käytettävyyteen suhtauduttu näissä samalla tavalla? Mitä eroja?